润乾
RAQSOFT

SPL程序设计

www.raqsoft.com.cn

目录

第1章 程序设计的基础概念 12

1.1 计算机程序和语言 13

1.2 集算器的安装 19

1.3 最简单的程序 24

1.4 更多简单程序举例 30

1.5 程序的结构 42

1.6 程序设计的任务 46

1.7 习题 48

第2章 算法与程序 54

2.1 什么是算法 55

2.2 简单的算法举例 59

2.3 算法的特性 74

2.4 表示算法的方法 76

用流程图表示算法 78

用N-S流程图 表示算法 91

用伪代码表示算法 94

用SPL表示算法 96

2.5 结构化的程序设计方法 100

2.6 习题 102

第3章 顺序结构程序设计 111

3.1 顺序程序设计举例 112

3.2 数据的表现形式及其运算 119

常量和变量 120

数据类型 124

数据类型的识别 134

3.3 运算符和表达式 136

SPL中常用的运算符 137

基本的算术运算符 139

运算符的优先级 140
多类型数据混合运算 141
强制类型转换运算符 143
3.4 SPL语句 144
3.5 数据的输入输出 158
输入输出问题举例 159
在SPL中输入输出数据 161
3.6 习题 162
第4章 选择结构程序设计 170
4.1 选择结构和条件判断 171
4.2 用 if 语句实现选择结构 173
选择结构举例 174
If 语句的一般形式 182
If 语句的其它形式 184
4.3 关系运算符和关系表达式 186
关系运算符及优先次序 187
关系表达式 189
4.4 逻辑运算符和逻辑表达式 190
逻辑运算符及优先次序 191
逻辑表达式 193
4.5 if 函数 196
4.6 选择结构的嵌套 202
4.7 习题 205
第5章 循环结构程序设计 225
5.1 对循环结构的需求 226
5.2 用for x语句实现循环结构 228
for x语句的一般形式 229
指定次数的循环 234
break和next语句 240
5.3 指定循环范围的for语句 248

for a, b语句 249
for a, b, s语句 252
5.4 循环结构的嵌套 255
循环结构结构的嵌套 256
在多层循环中使用break和next 259
5.5 循环结构举例 262
5.6 习题 271
第6章 文本与字符串 287
6.1 字符串 288
字符串常数 289
字符串操作符 291
特殊的字符串 292
转义符与特殊字符 293
6.2 字符串的转换 296
字符与ASCII码 297
编码转换 301
字符串与显示格式 303
6.3 动态程序 310
宏 311
eval函数 312
6.4 文本文件 316
6.5 字符串函数 323
字符函数 324
字符串处理函数 326
正则表达式 330
6.6 习题 332
第7章 函数与子程序 348
7.1 什么是函数 350
7.2 在SPL中调用函数 352
函数的调用 353

SPL中的函数参数..... 357
SPL中的函数选项..... 359
7.3 子程序..... 362
子程序的定义及调用..... 363
子程序中参数的传递..... 366
变量的作用范围..... 367
子程序的调用过程..... 369
子程序的嵌套..... 370
子程序的递归..... 373
7.4 跨网格调用..... 374
跨网格调用..... 375
跨网格调用参数的传递..... 377
跨网格调用的嵌套和递归..... 379
7.5 习题..... 380
第8章 序列..... 394
8.1 序列的概念..... 395
8.2 序列的产生与修改..... 401
序列常数..... 402
序列的成员访问..... 409
在序列最后添加成员..... 418
8.3 序列的循环与计算..... 421
用 for A 执行循环..... 422
聚合..... 431
序列计算举例..... 434
8.4 常见的序列..... 439
数列..... 440
字符串序列..... 445
8.5 多维序列..... 452
什么是多维序列..... 453
多维序列成员的访问..... 455

多维序列的使用 457
8.6 习题 463
第9章 序列的计算 483
9.1 序列的集合性 484
序列的有序性 485
序列与集合 490
9.2 序列函数 506
序列的生成与维护 507
字符串与序列 519
9.3 聚合计算 525
计数、求和与均值 527
排名 535
极值和中位数 540
其它聚合函数 545
9.4 循环计算 547
用A.(x)完成循环计算 548
定位计算 556
循环函数 561
9.5 定位 570
查找指定成员的位置 572
查找满足条件成员的位置 575
查找最大/最小值的位置 578
整体定位 583
9.6 选出 601
查找指定成员的位置 603
最大/最小值选出 606
排序和排名 611
9.7 多维序列的计算 617
9.8 序列计算举例 623
9.9 习题 632

第10章 数据表 642
10.1 结构化数据 643
结构化数据 644
字段与记录 650
数据表 651
10.2 序表的操作 654
序表创建/查看 655
记录访问 657
基本增删改 661
举例与延伸操作 672
10.3 外部数据 679
数据文件 680
文件数据读取 682
文件数据写出 689
数据库查询 695
10.4 序表的运算 696
聚合 697
过滤 703
10.5 排列 706
排列概念 707
排列的运算 717
排列与序表的不同 729
10.6 习题 734
第11章 有关表的计算 744
11.1 聚合 745
11.2 过滤与定位 757
过滤 758
定位 761
定位计算 767
11.3 对位计算 769

修改 770
产生 773
附加 779
11.4 排序与排名 782
排序 783
排序定位 791
排名 793
11.5 极值问题 803
极值 804
TOP 811
11.6 习题 820
第12章 有序计算 832
12.1 有序集合 833
有序集合的概念 834
序号访问 836
12.2 迭代函数 851
12.3 定位计算 857
定位成员 858
定位最大值/最小值 863
获取满足条件的成员序号 867
12.4 排序与对齐 871
排序 872
12.5 习题 881
第13章 分组计算 897
13.1 分组汇总 898
分组汇总 899
计算唯一值 913
13.2 分组的本质 922
分组子集 923
分组汇总与分组 933

13.3 非等值分组 939
等值分组与非等值分组 940
对齐分组 942
枚举分组 950
13.4 有序分组 960
分组的顺序 961
有序分组 968
13.5 习题 978
第14章 表间关联 1000
14.1 主键与索引 1001
根据主键查找 1002
索引表 1007
14.2 表间关联 1010
表间关联 1011
常见的关联关系 1015
14.3 外键表 1021
14.4 同维表和主子表 1051
14.5 交叉连接与非等值连接 1068
交叉分组 1069
非等值连接 1073
14.6 习题 1077
第15章 图形绘制 1100
15.1 画布 1101
画布 1102
图元 1103
15.2 物理坐标 1107
绝对坐标 1113
相对坐标 1116
15.3 逻辑坐标 1119
数值轴 1123

枚举轴.....1132
日期轴.....1142
15.4 更多逻辑映射.....1145
极坐标系.....1146
组合坐标系.....1153
图例.....1157
15.5 成品图.....1169
柱状图.....1171
双轴图.....1173
15.6 习题.....1176
三角函数.....1177
多重组合.....1180
K线图.....1183
第16章 大数据与游标.....1186
16.1 大数据与游标.....1187
大数据.....1188
游标概念.....1191
16.2 基础游标操作.....1193
用游标读取数据.....1194
循环游标.....1200
16.3 游标计算函数.....1209
游标的聚合计算.....1210
使用游标函数.....1217
延迟计算与立即计算.....1232
16.4 有序游标.....1234
16.5 大游标.....1245
大排序.....1246
大分组.....1249
合并.....1252
归并.....1256

连接..... 1260

多路游标..... 1265

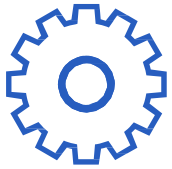
16.6 习题..... 1271

第1章 程序设计的基础概念

1.1 计算机程序和语言
1.2 集算器的安装
1.3 最简单的程序
1.4 更多简单程序举例
1.5 程序的结构
1.6 程序设计的任务
1.7 习题

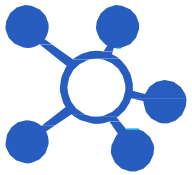
1.1 计算机程序和语言

计 算 机 程 序



指 令

指示计算机执行某种基本操作的命令



程 序

一组指令的集合

一个特定的序列执行，用来完成特定的功能

软 件

一系列按照特定顺序组织的程序和可能需要的数据及文件等的集合

程序设计的最终结果

计 算 机 语 言

	优点	缺点
机器语言	计算机直接识别 执行效率最高	可读性极差，难学习，难检查，难修改 依赖计算机硬件，难以迁移
汇编语言	比机器语言的可读性好 执行效率高	可读性比较差，代码冗长，不易修改 同样依赖计算机硬件，难以迁移
高级语言	简单易学，易于理解，容易修改 功能强大，可移植	执行效率较低 有各种不同用途，概念语法各不相同

1.1 计算机程序和语言 - 计算机语言

高级语言

1000+100

高级语言易读易写，人们很容易了解其意义

汇编语言

MOV AX,1000
ADD AX,100

汇编语言将高级语言转换为机器指令，对于不同的指令集，汇编语言也不同

机器语言

10111000
11101000
00000011
00000101
01100100
00000000

机器语言是计算机使用的二进制语言，不同架构的平台中机器语言并不通用

编译和解释

编 译

- **编译**（Compilation）是将高级语言**一次性**生成为计算机能够识别的机器语言的过程。
- 执行编译过程的程序叫做**编译器**，程序需要先编译后才能执行。
- 如果需要在不同平台上使用，或者程序有变化，需要**重新编译**后再执行。
- 使用编译执行的语言称为**静态语言**，如C/C++,Pascal,Fortran等。

解 释

- **解释**（Interpretation）是将高级语言逐条转换为目标代码执行的过程。
- 执行解释过程的程序叫做**解释器**，程序边解释边执行。
- 解释执行的程序，移植或修改时不需重新编译。
- 使用解释执行的语言称为**脚本语言**，如SPL， Python， JavaScript等。

高级语言的发展

早期计算机语言
跳转多，缺乏规范
开发效率低，难维护

程序具有良好特性的基本结构，
不允许随意跳转
程序结构清晰，易编写易维护

以对象作为基本结构
提供封装、继承和重载等特性
拥有更强的可复用能力

上面这些内容不理解也没有关系，只是简单了解术语，不影响后面的学习

1.2 集算器的安装

为什么需要集算器?

- 解决复杂的实际问题，需要一种简单的高级语言
- 经常变化的运算逻辑和数据，需要一种解释执行的语言，以避免反复重新编译
- 业务中频繁需要关联计算和统计分析，需要一种用于结构化数据计算的工具

SPL是什么?

- SPL(Structured Process Language)是集算器采用的结构化计算语言
- SPL是一种脚本语言，边解释边执行，无需提前编译
- SPL是一种语法更接近自然思维的、更易掌握的高级语言

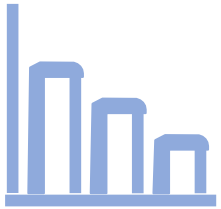
SPL(Structured Process Language))

解释执行的语言

边解释边执行，效率较高
代码简洁，易读易写
使用方便，易维护

结构化查询

数据结构清晰
支持多类型数据
交互式查询，具备高度的灵活性



高效数据分析

计算功能强大
智能化数据处理
支持并行计算，进一步提高性能

网格化代码

编程简单快捷
方便的网格变量
分步式执行，结果可见

集算器的下载和安装

https://www.raqsoft.com.cn/download/desktop-download

1.3 最简单的程序

【例1.1】要求在控制窗口中输出：Hello, world!

【例1.1】要求在控制窗口中输出：Hello, world!

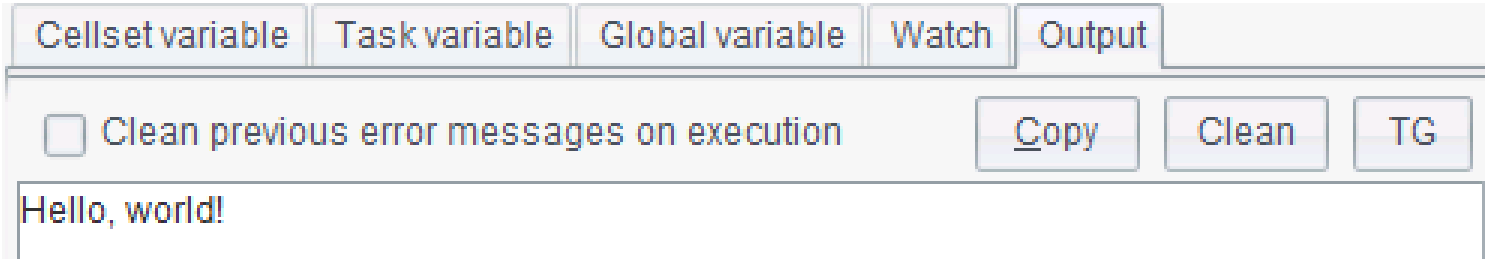
解题思路: 在单元格中用output函数原样输出以上字符串。

按 按钮创建网格文件

	A	B
1	>output("Hello, world!")	/输出一行文字

执行方法: 无需编译，在上方工具栏中按 执行。

输出结果: 在右侧的控制台窗口中可以看到代码执行结果。



1.3 最简单的程序

	A	B
1	>output("Hello, world!")	/输出一行文字

程序分析

- SPL代码写在一个**网格**中，存储为.dfx文件。
- 每条语句都写在一个**单元格**中。
- 以>开头的单元格为**执行格**，可用来执行计算或操作。
- 以 /开头的单元格为**注释格**，可以用来对程序进行必要的说明，以便于维护代码。

1.3 最简单的程序

	A	B
1	>output("Hello, world!")	/输出一行文字

程序分析

- **output()**是SPL中的输出函数，用来在控制台输出一行文字。output函数中双引号内的字符串Hello, world!将按原样输出。

- **注释格不参与计算**，对程序的运行不起作用。注释格中的文字会变为**绿色**，仅用来给人查看。

1.3 最简单的程序

	A	B
1	>output("Hello, world!")	/输出一行文字

程序分析

- SPL程序**无需编译**，可以直接执行。以dfx扩展名存储的网格文件也可以在其它程序中调用（详见第6章函数与子程序）
- SPL语句写在单元格中，执行时通常按照**从上到下，从左到右**的顺序执行。

1.4 更多简单程序举例

【例1.2】根据给定的身高(m)和体重(kg)计算BMI指数。

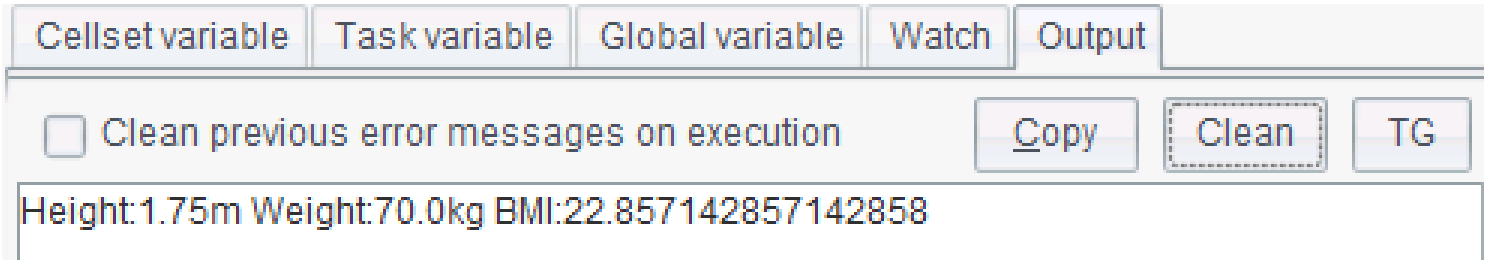
1.4 更多简单程序举例

【例1.2】根据给定的身高(m)和体重(kg)计算BMI指数。

解题思路: 根据单元格中给出的身高和体重数据计算，并输出结果。

	A	B
1	1.75	/身高，m
2	70.0	/体重，kg
3	=A2/A1/A1	/计算BMI指数
4	>output("Height:"/A1/"m Weight:"/A2/"kg BMI:"/A3)	/输出结果

输出结果:



1.4 更多简单程序举例

【例1.2】根据给定的身高(m)和体重(kg)计算BMI指数。

	A	B
1	1.75	/身高，m
2	70.0	/体重，kg
3	=A2/A1/A1	/计算BMI指数
4	>output("Height:"/A1/"m Weight:"/A2/"kg BMI:"/A3)	/输出结果

程序分析

- SPL中，数据可以直接记录在单元格中，如A1，A2格，这样的单元格称为**常数单元格**，格子中的数据显示为洋红色。
- 计算时可以直接用单元格名引用常数单元格中的数据。
- 以=开头的单元格为**计算格**，可用来执行计算，并返回结果。

1.4 更多简单程序举例

【例1.2】根据给定的身高(m)和体重(kg)计算BMI指数。

	A	B
1	1.75	/身高，m
2	70.0	/体重，kg
3	=A2/A1/A1	/计算BMI指数
4	>output("Height:"/A1/"m Weight:"/A2/"kg BMI:"/A3)	/输出结果

- **常数**又称为常量，是一个固定不变的值/量，并且有着一定含义的名称。上面例子中使用的常数格的格值就是一种常数，它的名称就是格子位置的名称，如A1、A2，常数格的初始格值就是其中数据的值，如1.75和70.0，常数格的值通常不会被修改。
- **计算**是指根据使用的表达式计算出结果，表达式中的常数(或变量)需根据名称转换为对应的值，如A3中表达式为=A2/A1/A1，计算A3就是得到70/1.75/1.75的结果。
- **变量**又称为变数，是有着一定含义的名称，并代表着某个可能变化的数值。SPL中的单元格全部可以作为变量来使用，变量值就是单元格的格值，它们的名称则是格子位置对应的名称。

有关常数和变量等概念，将在第3章中详细讲解，这里只需初步了解。

1.4 更多简单程序举例

【例1.2】根据给定的身高(m)和体重(kg)计算BMI指数。

	A	B
1	1.75	/身高，m
2	70.0	/体重，kg
3	=A2/A1/A1	/计算BMI指数
4	>output("Height:"/A1/"m Weight:"/A2/"kg BMI:"/A3)	/输出结果

程序分析

- output() 函数输出的字符串由多个部分拼成，字符串拼接可以用操作符/，而同样的操作符在数值运算中表示除法。

1.4 更多简单程序举例

【例1.2】根据给定的身高(m)和体重(kg)计算BMI指数。

	A	B
1	1.75	/身高，m
2	70.0	/体重，kg
3	=A2/A1/A1	/计算BMI指数
4	>output("Height:"/A1/"m Weight:"/A2/"kg BMI:"/A3)	/输出结果

程序分析

- 常数格的格值就是格子中的常数，也可以选中单元格查看：

Value
1.75

Value
70.0

- 执行格、注释格或者空白单元格等是没有格值的：

Value
(null)

1.4 更多简单程序举例

【例1.2】根据给定的身高(m)和体重(kg)计算BMI指数。

	A	B
1	1.75	/身高，m
2	70.0	/体重，kg
3	=A2/A1/A1	/计算BMI指数
4	>output("Height:"/A1/"m Weight:"/A2/"kg BMI:"/A3)	/输出结果

	A	B
1	1.75	70.0
2	=B1/A1/A1	>output("Height:"/A1/"m Weight:"/B1/"kg BMI:"/A2)

程序分析

- SPL程序可以写在多列单元格中，网格执行时的顺序是从上到下，从左到右，上面两个程序的效果是相同的。

1.4 更多简单程序举例

【例1.3】输入3个数，求最大值。

1.4 更多简单程序举例

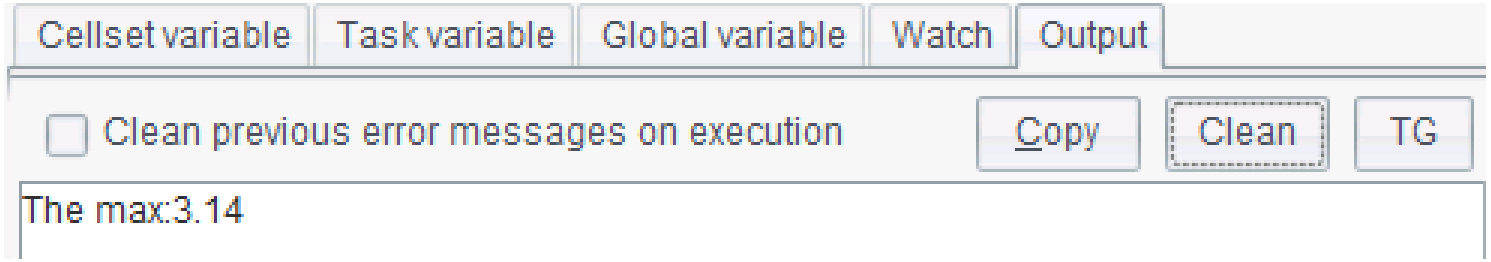
【例1.3】输入3个数，求最大值。

解题思路: 在网格中用常数格输入3个数，通过比较求出最大值并输出。

在程序中比较3个单元格常数，获得最大值并输出：

	A	B	C	D
1	3	3.14	3.139889	/用常数格输入3个数
2	=if(A1>B1, A1, B1)			/获得A1和B1中的较大值
3	=if(C1>A2, C1, A2)			/获得C1和A2中的较大值
4	>output("The max:"/A3)			/A3中即为最大值，输出结果

输出结果:



1.4 更多简单程序举例

【例1.3】输入3个数，求最大值。

	A	B	C	D
1	3	3.14	3.139889	/用常数格输入3个数
2	=if(A1>B1, A1, B1)			/获得A1和B1中的较大值
3	=if(C1>A2, C1, A2)			/获得C1和A2中的较大值
4	>output(“The max:"/A3)			/A3中即为最大值，输出结果

程序分析

- if() 函数有3个参数，参数之间用**逗号**分隔，第1个参数为条件，如果成立则返回第2个参数，否则返回第3个参数。在A1中，如果网格参数A1>B1则返回A1，否则返回B1。有关函数的参数，将在下一节中讲解。
- SPL中可以访问前面计算过的格值，分步完成计算。

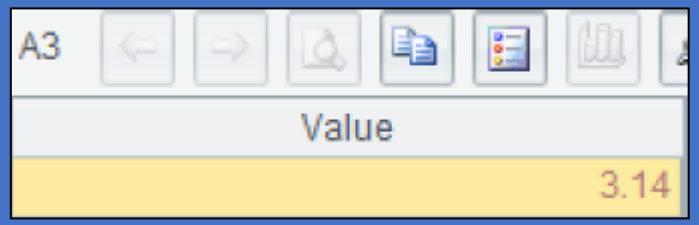
1.4 更多简单程序举例

【例1.3】输入3个数，求最大值。

	A	B	C	D
1	3	3.14	3.139889	/用常数格输入3个数
2	=if(A1>B1, A1, B1)			/获得A1和B1中的较大值
3	=if(C1>A2, C1, A2)			/获得C1和A2中的较大值

程序分析

- 用SPL执行计算时，也可以不用output函数输出结果，而是在计算后查看格值。如上面的程序中，计算后，A3的格值即为3个数的最大值：



因此，SPL程序不一定需要将结果用output函数输出，也可以在计算完成后查看格值。

1.5 程序的结构

1.5 程序的结构

SPL程序写在一个网格中，由各个单元格中的语句构成。

网格中可以用常数格输入数据，可以用注释格记录说明文字，可以用计算格获得计算结果，也可以用执行格完成操作或计算。

其它高级语言也是类似的，由执行语句和注释语句构成，按一定次序执行

网格程序一般按照从上到下，从左到右的顺序执行

计算时可使用单元格的名称引用它的格值作为常数或变量使用。

SPL可输入网格参数，计算结果可存储在单元格值中，也可用函数输出

程序中应该包含必要的注释

常数格

计算格

执行格

注释格

	A	B
1	1.75	/身高，m
2	70.0	/体重，kg
3	=A2/A1/A1	/计算BMI指数
4	>output("Height:"/A1/"m Weight:"/A2/"kg BMI:"/A3)	/输出结果

函数调用

	A	B	C	D
1	3	3.14	3.139889	/用常数格输入3个数
2	=if(A1>B1, A1, B1)			/获得A1和B1中的较大值
3	=if(C1>A2, C1, A2)			/获得C1和A2中的较大值

= if (C1>A2, C1, A2)

函数名　　参数

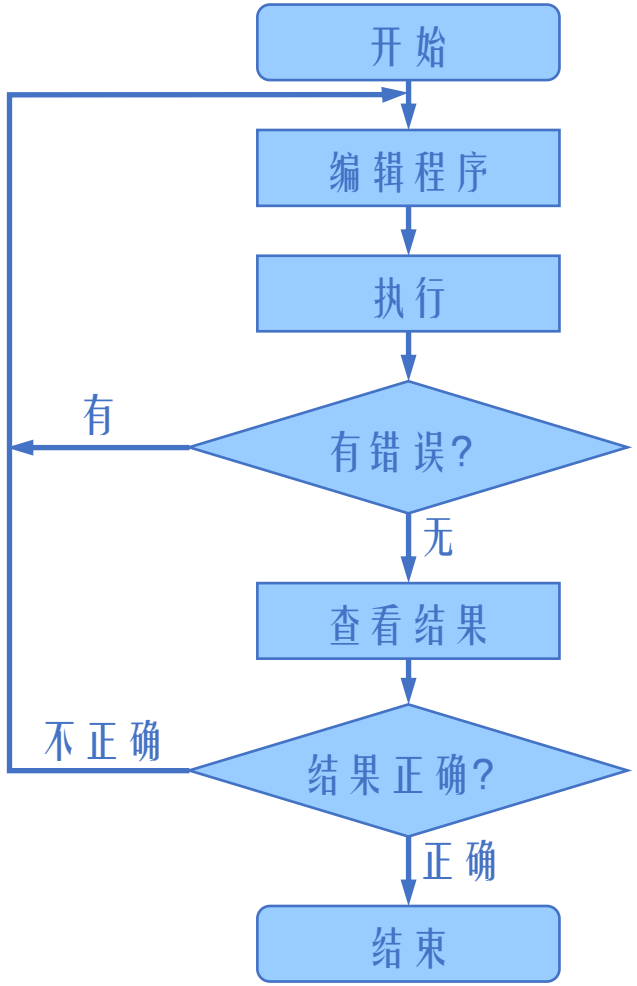
参数可以是数值、字符串、单元格名、网格参数名等，也可能是表达式。

SPL语法

if函数还可以使用多组条件参数，如：
=if(A1>B1 && A1>C1, A1, B1>A1 && B1>C1, B1, C1)
执行时，当A1中的值大于B1也大于C1时，返回结果为A1；否则，当B1中的值大于A1也大于C1时，返回结果为B1；如果前面都不满足，则返回结果C1。

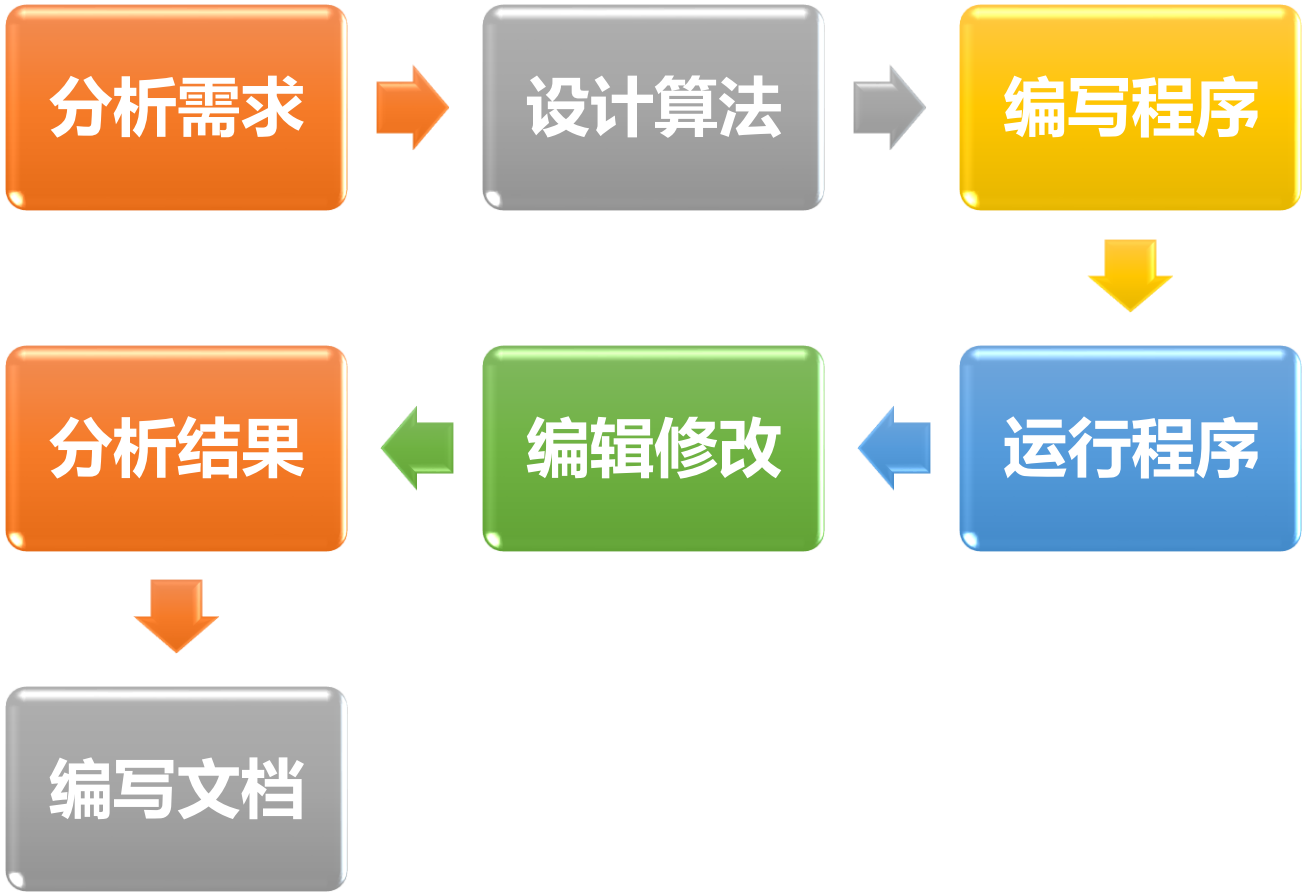
&&为逻辑运算符与，表示左右两侧的条件全部成立。
有关if函数和逻辑运算符，将在第4章中详细介绍。

程序编写步骤



1.6 程序设计的任务

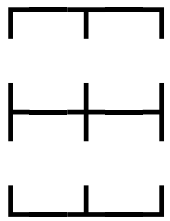
1.6 程序设计的任务



1.7 习题

1.7 习题

【习题1.1】写程序输出下面的网格：



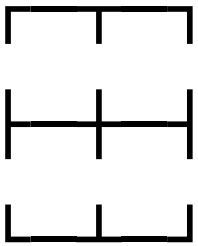
【习题1.2】写定两个英文姓名，用来输出"Hello, xxx. I'm xxx." 。

【习题1.3】在网格中直接写入π为3.1416，和圆的半径r ，算出圆的周长和面积。

【习题1.4】输入2个数，输出它们的大小关系，是a>b, a<b还是a==b。

1.7 习题

【习题1.1】写程序输出下面的网格：



	A	B
1	>output(" ┌─┬─┐ ")	/逐行输出表格字符，输出第一行
2	>output(" ├─┼─┤ ")	/输出第二行
3	>output(" └─┴─┘ ")	/输出第三行

1.7 习题

【习题1.2】写定两个英文姓名，用来输出"Hello, xxx. I'm xxx." 。

	A	B
1	Tom	/用常数格记录两个英文姓名
2	Jerry	
3	>output("Hello, "/A1/". I'm "/A2/".")	/output函数的参数中，用"/"操作符连接字符串

1.7 习题

【习题1.3】在网格中直接写入π为3.1416，和圆的半径r ， 算出圆的周长和面积。

	A	B	C
1	3.1416	4	/用常数格输入π和r
2	=2*A1*B1		/计算圆周长
3	=A1*B1*B1		/计算圆面积
4	>output("r="/B1/", circumference="/A2/", area="/A3)		/输出结果

1.7 习题

【习题1.4】输入2个数，输出它们的大小关系，是a>b, a<b还是a==b。

	A	B	C
1	4	4.789	/用常数格输入2个数
2	=if(A1>B1, ">", A1<B1, "<", "==")		/if函数中可以连续使用几个条件
3	>output(A1/A2/B1)		/输出结果

第2章 算法与程序

2.1 什么是算法

2.2 简单的算法举例

2.3 算法的特性

2.4 表示算法的方法

- 用流程图表示算法
- 用N-S流程图 表示算法
- 用伪代码表示算法
- 用SPL表示算法

2.5 结构化的程序设计方法

2.6 习题

2.1 什么是算法

2.1 什么是算法

尼克劳斯 沃斯
(Niklaus Wirth)

瑞士计算机科学家
图灵奖获得者
Pascal之父
多种编程语言设计者

算法+数据结构=程序

数据结构

指计算机存储和使用数据的方式。
在程序中使用到的各种数据的描述，
以及它们的类型和组织形式。

算法

指计算机解决问题的操作描述。
程序求解特定问题的流程和步骤，
是有限条计算机指令的集合。

算法

为解决特定问题而采取的流程和步骤就是算法。

算法是对操作方案完整而准确的描述，在程序中，算法是一系列明确清晰的指令。

为了解决同一个问题，可以采用不同的算法。

为了更有效地解决问题，不仅需要算法能够得到正确的结果，也需要考虑算法的成本、效率等因素，选择更合适的算法。

2.1 什么是算法

数值运算算法	非数值运算算法
目的是求数值解	目的各异，难以规范化
往往可以运用已有数值分析方法	算法往往需要自行设计
算法比较成熟	算法差异大，同一问题可有不同算法

2.2 简单的算法举例

【例2.1】计算5×5×5×5×5的结果。

2.2 简单的算法举例

【例2.1】计算5×5×5×5×5的结果。

算法步骤

Step1: 先求5乘以5，得到结果25

Step2: 将Step1的结果再乘以5，得到结果125

Step3: 上一步结果125再乘以5，得到结果625

Step4: 将625乘以5，得到最终结果**3125**

2.2 简单的算法举例

【例2.1】计算5×5×5×5×5的结果。

算法步骤

Step1: 先求5乘以5，得到结果25

Step2: 将Step1的结果再乘以5，得到结果125

Step3: 上一步结果125再乘以5，得到结果625

Step4: 将625乘以5，得到最终结果**3125**

算法步骤

Step1: 令 p=5, 或写作 p←5(表示将 p 赋值为5)

Step2: 令 i=1, 或写作 i←1(表示将 i 赋值为1)

Step3: 使p与5相乘，结果仍然存储在变量p中，或写作p←p*5

Step4: 使 i 的值加1, 可写作 i←i+1

Step5: 若 i<5，返回执行Step3，并执行其后的步骤；否则结束。

2.2 简单的算法举例

【例2.1】计算5×5×5×5×5的结果。

算法步骤

Step1: 先求5乘以5，得到结果25

Step2: 将Step1的结果再乘以5，得到结果125

Step3: 上一步结果125再乘以5，得到结果625

Step4: 将625乘以5，得到最终结果**3125**

题目修改为计算9×9×9×9×9×9的结果。

算法步骤

Step1: 令 p=9, 或写作 p←9(表示将 p 赋值为9)

Step2: 令 i=1, 或写作 i←1(表示将 i 赋值为1)

Step3: 使p与9相乘，结果仍然存储在变量p中，或写作p←p*9

Step4: 使 i 的值加1, 可写作 i←i+1

Step5: 若 i<6，返回执行Step3，并执行其后的步骤；否则结束。

这种算法更为灵活，能用来解决类似问题，具有一般性和通用性

【例2.2】欧洲有44个国家，列出首都的首字母为“S”的国家和首都的名称。

2.2 简单的算法举例

【例2.2】欧洲有44个国家，列出首都的首字母为“S”的国家和首都的名称。

N：　表示国家名称

下标i：　表示第几个国家的名称

N_1：　表示第1个国家的名称

N_i：　表示第 i 个国家的名称

C：　表示首都名称

C_1：　表示第1个国家的首都名称

C_i：　表示第 i 个国家的首都名称

算法步骤

Step1: i←1

Step2: 若 C_i 的首字母是S，输出N_i和C_i；否则不输出。

Step3: i←i+1

Step4: 若 i≤44，返回到Step2，继续执行；否则结束。

【例2.3】根据泰勒公式计算：$\pi \approx 4-\frac{4}{3}+\frac{4}{5}-\frac{4}{7}+\frac{4}{9}-\frac{4}{11}+\cdots+\frac{4}{197}-\frac{4}{199}$

2.2 简单的算法举例

【例2.3】根据泰勒公式计算：$\pi \approx 4-\frac{4}{3}+\frac{4}{5}-\frac{4}{7}+\frac{4}{9}-\frac{4}{11}+\cdots+\frac{4}{197}-\frac{4}{199}$

term： 累加时其中的一项

sign： 当前项的正负号，1或-1

deno： 当前项的分母

res： 当前的累加结果

算法步骤

Step1: res←4

Step2: sign←-1

Step3: deno←3

Step4: term←sign*(4/deno)

Step5: res←res+term

Step6: sign←sign*(-1)

Step7: deno←deno+2

Step8: 若 deno≤199，返回到Step4，继续执行；否则结束并输出res。

【例2.4】请列出2019年之后的连续30个闰年。

【例2.4】请列出2019年之后的连续30个闰年。

【例2.4】请列出2019年之后的连续30个闰年。

算法步骤

Step1: year←2019

Step2: num←0

Step3: 若year不能被4整除，则这一年不是闰年，转到Step7继续执行，检查下一年的情况。

Step4: 若year不能被100整除，则这一年是闰年，转到Step6继续执行；否则继续判断。

Step5: 若year能被400整除，则这一年是闰年，继续执行；否则说明不是闰年，跳转到Step7。

Step6: 找到一个闰年，输出year，num←num+1

Step7: year←year+1

Step8: 若 num<30，转到Step3继续执行，判断下一年；否则寻找完成，算法结束。

【例2.5】给出一个正整数，输出其质因数分解等式，如105=3*5*7。

2.2 简单的算法举例

【例2.5】给出一个正整数，输出其质因数分解等式，如105=3*5*7。

- 将一个正整数分解质因数，只需要从2开始依次寻找它的约数即可
- 如果正整数n无法被不大于$\sqrt{n}$的任何一个质数整除，那么它一定是质数
- 如果正整数n≤3，是无法进一步分解的

2.2 简单的算法举例

【例2.5】给出一个正整数，输出其质因数分解等式，如105=3*5*7。

- 将一个正整数分解质因数，只需要从2开始依次寻找它的约数即可
- 如果正整数n无法被不大于$\sqrt{n}$的任何一个质数整除，那么它一定是质数
- 如果正整数n≤3，是无法进一步分解的

算法步骤

Step1: 输入n的值

Step2: i←2 ， s←n+"="

Step3: 若 n≤3，说明分解完成，执行Step8。

Step4: 计算n除以i的余数r

Step5: 若 r=0，说明n能够被i整除，则n←n/i，s←s+i+"*"，返回到Step3，继续执行。

Step6: i←i+1

Step7: 若 i≤$\sqrt{n}$，返回到Step4继续执行。

Step8: s←s+n，输出s并结束算法。

2.3 算法的特性

算法的特性

有穷性

一个算法的步骤是有限的，在执行后能够终止。

确定性

算法的每一个步骤都必须有确切的定义，不能有歧义。

输入项

算法在可以从外界获取0到多个输入信息作为初始条件。0个输入时是指算法已定义了初始条件。

输出项

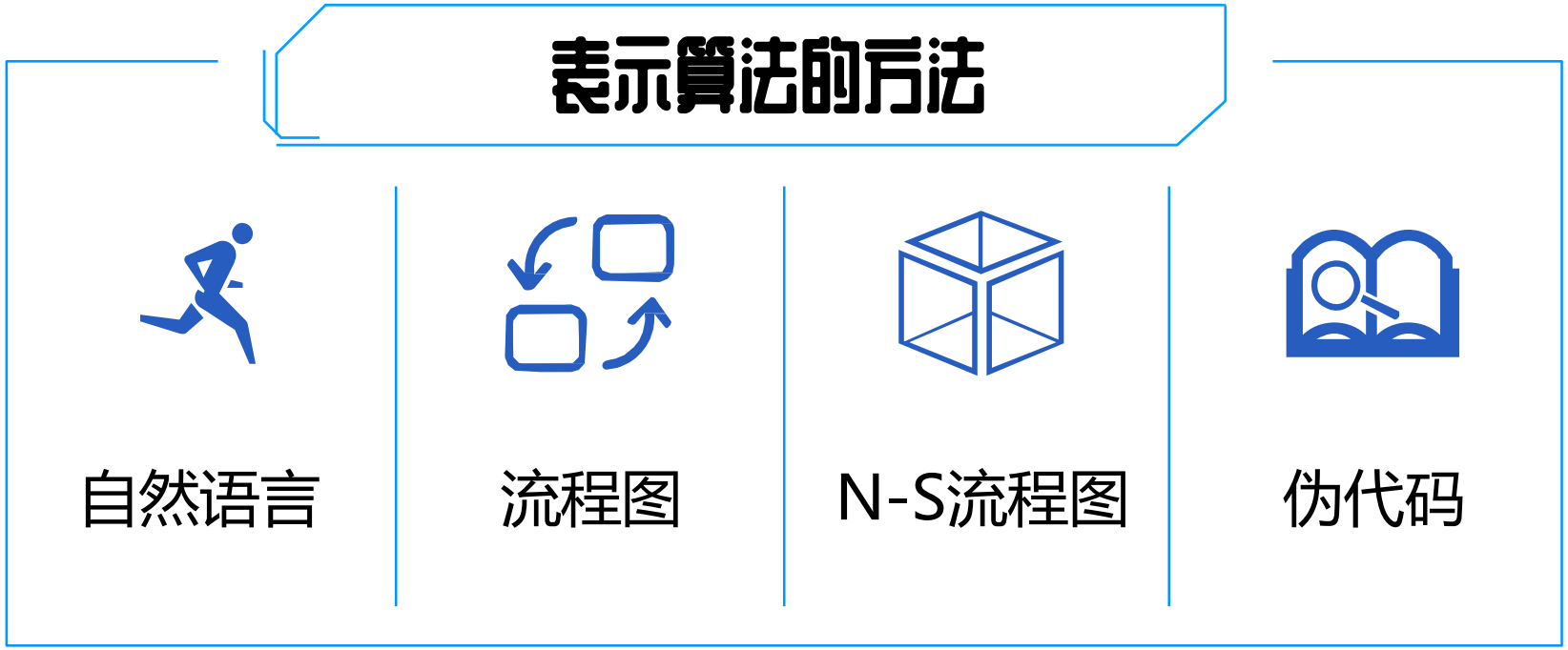
一个算法需要有1个或多个输出，用来返回结果。

可行性

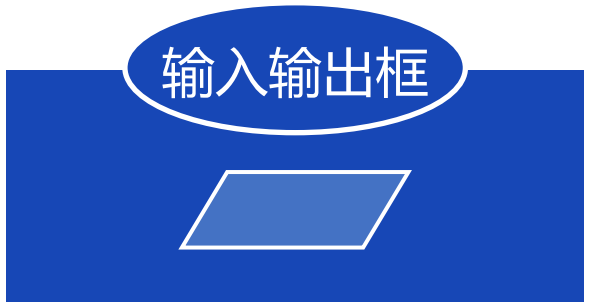
算法中的每一个步骤都是能够在有限的时间内有效地执行，并得到确切结果的。也称为有效性。

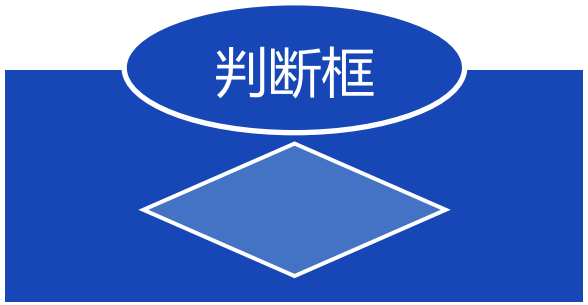
2.4 表示算法的方法

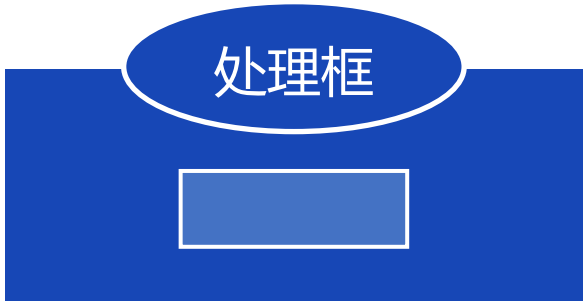
2.4 表示算法的方法

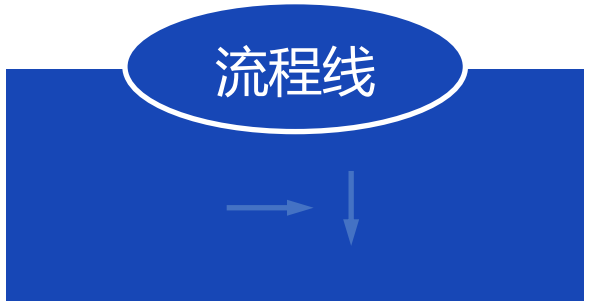


流程图的符号









【例2.6】将例2.1的算法用流程图表示。
计算5×5×5×5×5的结果。

算法步骤

Step1: p←5

Step2: i←1

Step3: p←p*5

Step4: i←i+1

Step5: 若 i≥5，输出p结束；否则返回Step3继续执行。

算法的流程图表示

【例2.6】将例2.1的算法用流程图表示。
计算5×5×5×5×5的结果。

算法步骤

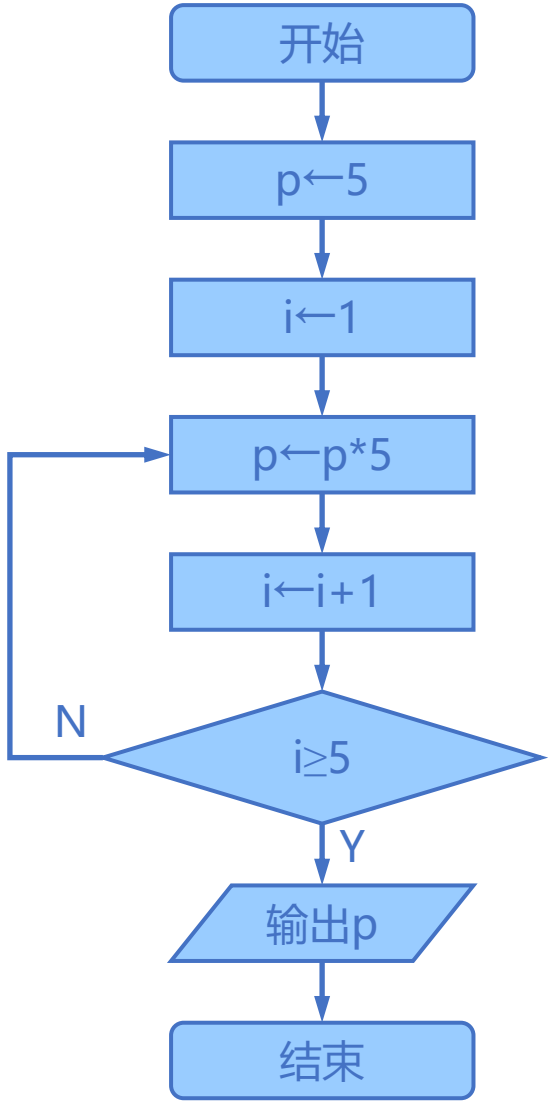
Step1: p←5

Step2: i←1

Step3: p←p*5

Step4: i←i+1

Step5: 若 i≥5，输出p结束；否则返回Step3继续执行。



【例2.7】将例2.2的算法用流程图表示。

欧洲有44个国家，列出首都的首字母为“S”的国家和首都的名称。

算法步骤

Step1: i←1

Step2: 若 C_i 的首字母是S，输出N_i 和C_i；否则不输出。

Step3: i←i+1

Step4: 若 i≤44，返回到Step2，继续执行；否则结束。

2.4 表示算法的方法

算法的流程图表示

【例2.7】将例2.2的算法用流程图表示。

欧洲有44个国家，列出首都的首字母为“S”的国家和首都的名称。

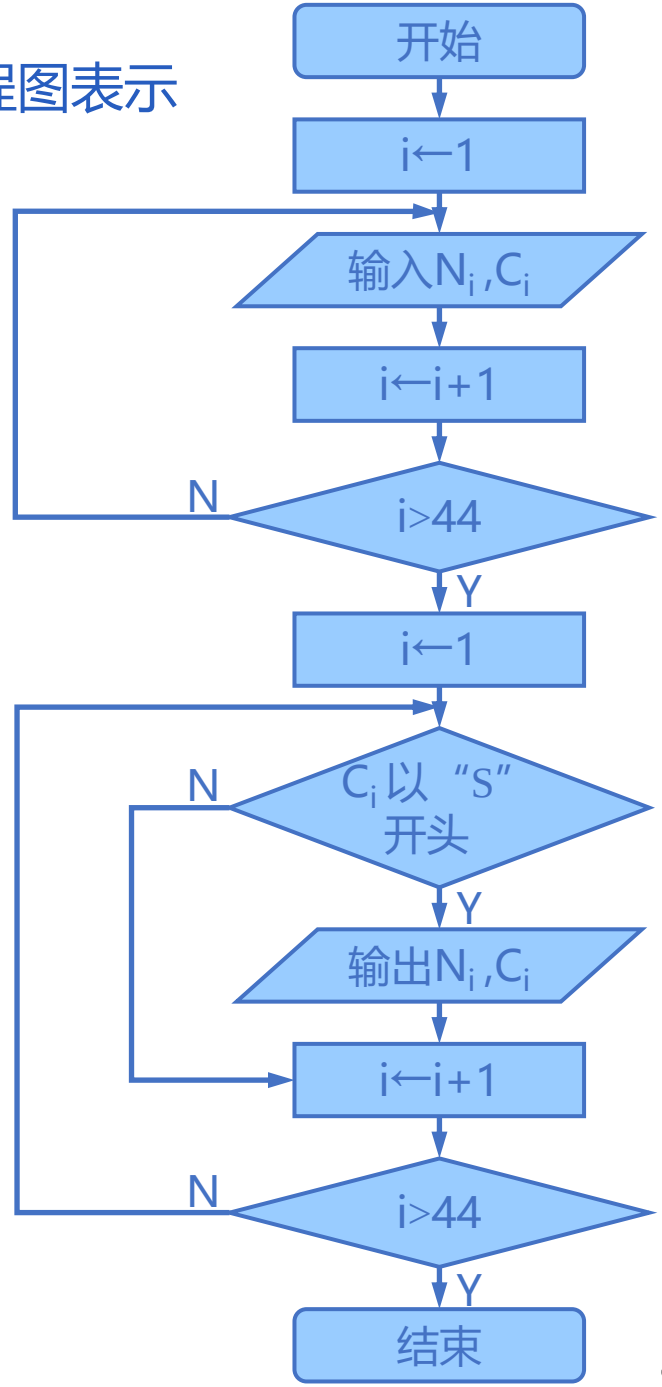
算法步骤

Step1: i←1

Step2: 若 C_i 的首字母是S，输出N_i 和C_i；否则不输出。

Step3: i←i+1

Step4: 若 i≤44，返回到Step2，继续执行；否则结束。



【例2.8】将例2.3的算法用流程图表示。

$$\text{计算 } \pi \approx 4-\frac{4}{3}+\frac{4}{5}-\frac{4}{7}+\frac{4}{9}-\frac{4}{11}+\cdots+\frac{4}{197}-\frac{4}{199}$$

算法步骤

Step1: res←4

Step2: sign←-1

Step3: deno←3

Step4: term←sign*(4/deno)

Step5: res←res+term

Step6: sign←sign*(-1)

Step7: deno←deno+2

Step8: 若 deno≤199，返回到Step4，继续执行；否则结束并输出res。

2.4 表示算法的方法

【例2.8】将例2.3的算法用流程图表示。

计算 $\pi \approx 4-\frac{4}{3}+\frac{4}{5}-\frac{4}{7}+\frac{4}{9}-\frac{4}{11}+\cdots+\frac{4}{197}-\frac{4}{199}$

算法步骤

Step1: res←4

Step2: sign←-1

Step3: deno←3

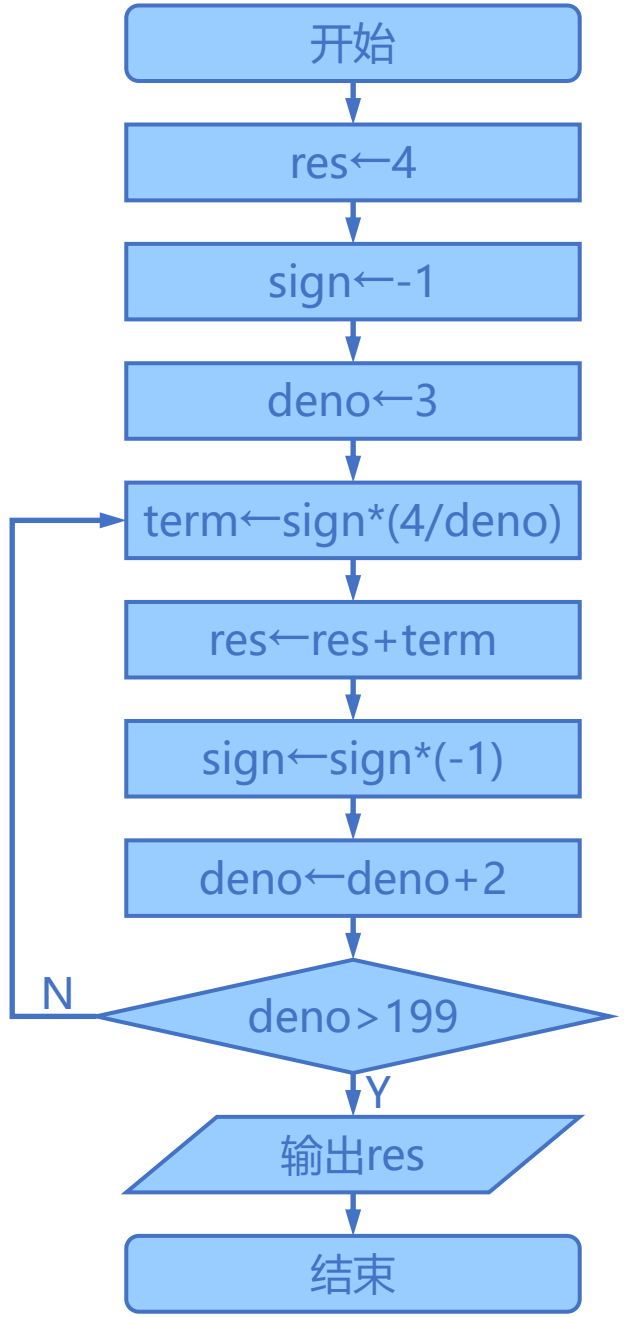
Step4: term←sign*(4/deno)

Step5: res←res+term

Step6: sign←sign*(-1)

Step7: deno←deno+2

Step8: 若 deno≤199，返回到Step4，继续执行；否则结束并输出res。



【例2.9】将例2.4的算法用流程图表示。
请列出2019年之后的连续30个闰年。

算法步骤

Step1: year←2019

Step2: num←0

Step3: 若year不能被4整除，则这一年不是闰年，转到Step7继续执行，检查下一年的情况。

Step4: 若year不能被100整除，则这一年是闰年，转到Step6继续执行；否则继续判断。

Step5: 若year能被400整除，则这一年是闰年，继续执行；否则说明不是闰年，跳转到Step7。

Step6: 找到一个闰年，输出year，num←num+1

Step7: year←year+1

Step8: 若 num<30，转到Step3继续执行，判断下一年；否则寻找完成，算法结束。

2.4 表示算法的方法

【例2.9】将例2.4的算法用流程图表示。
请列出2019年之后的连续30个闰年。

算法步骤

Step1: year←2019

Step2: num←0

Step3: 若year不能被4整除，则这一年不是闰年，转到Step7继续执行，检查下一年的情况。

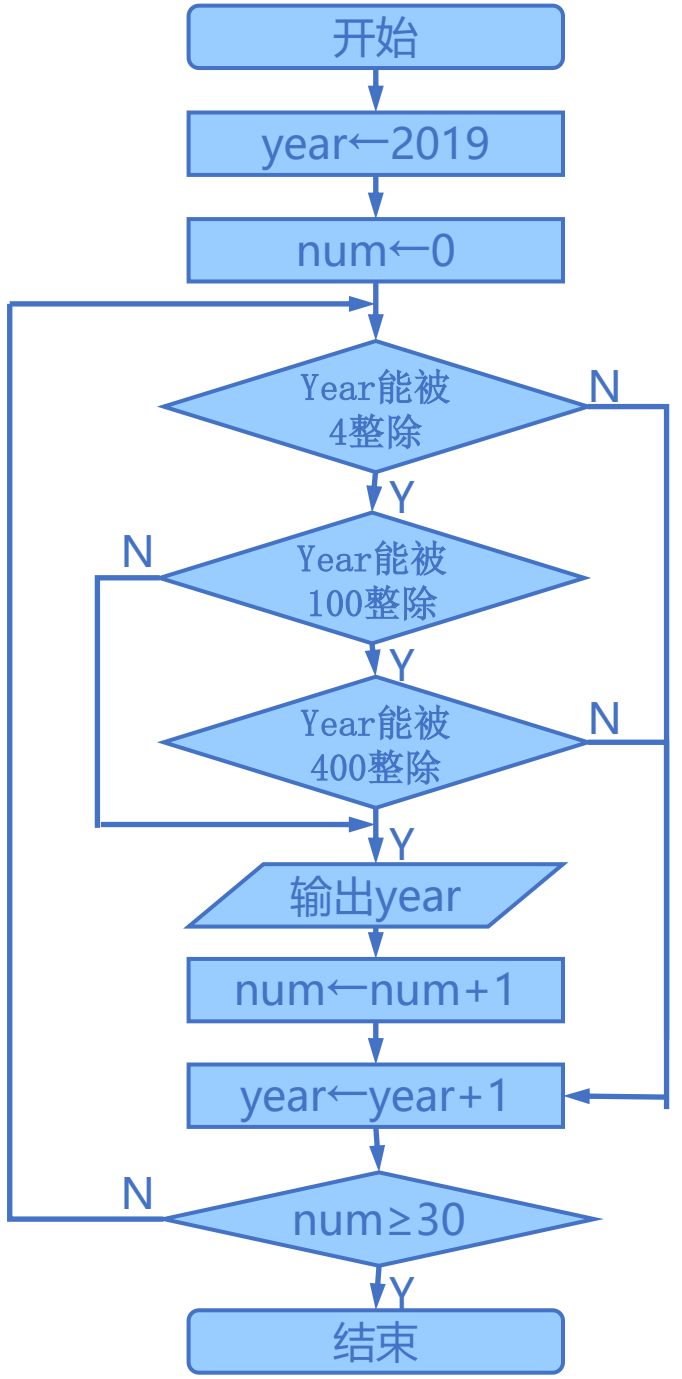
Step4: 若year不能被100整除，则这一年是闰年，转到Step6继续执行；否则继续判断。

Step5: 若year能被400整除，则这一年是闰年，继续执行；否则说明不是闰年，跳转到Step7。

Step6: 找到一个闰年，输出year，num←num+1

Step7: year←year+1

Step8: 若 num<30，转到Step3继续执行，判断下一年；否则寻找完成，算法结束。



【例2.10】将例2.5的算法用流程图表示。

给出一个正整数，输出其质因数分解等式。

算法步骤

Step1: 输入n的值

Step2: i←2 ， s←n+"="

Step3: 若 n≤3，说明分解完成，执行Step8。

Step4: 计算n除以i的余数r

Step5: 若 r=0，说明n能够被i整除，则n←n/i，s←s+i+"*"，返回到Step3，继续执行。

Step6: i←i+1

Step7: 若 i≤$\sqrt{n}$，返回到Step4继续执行。

Step8: s←s+n，输出s并结束算法。

2.4 表示算法的方法

【例2.10】将例2.5的算法用流程图表示。

给出一个正整数，输出其质因数分解等式。

算法步骤

Step1: 输入n的值

Step2: i←2 ， s←n+"="

Step3: 若 n≤3，说明分解完成，执行Step8。

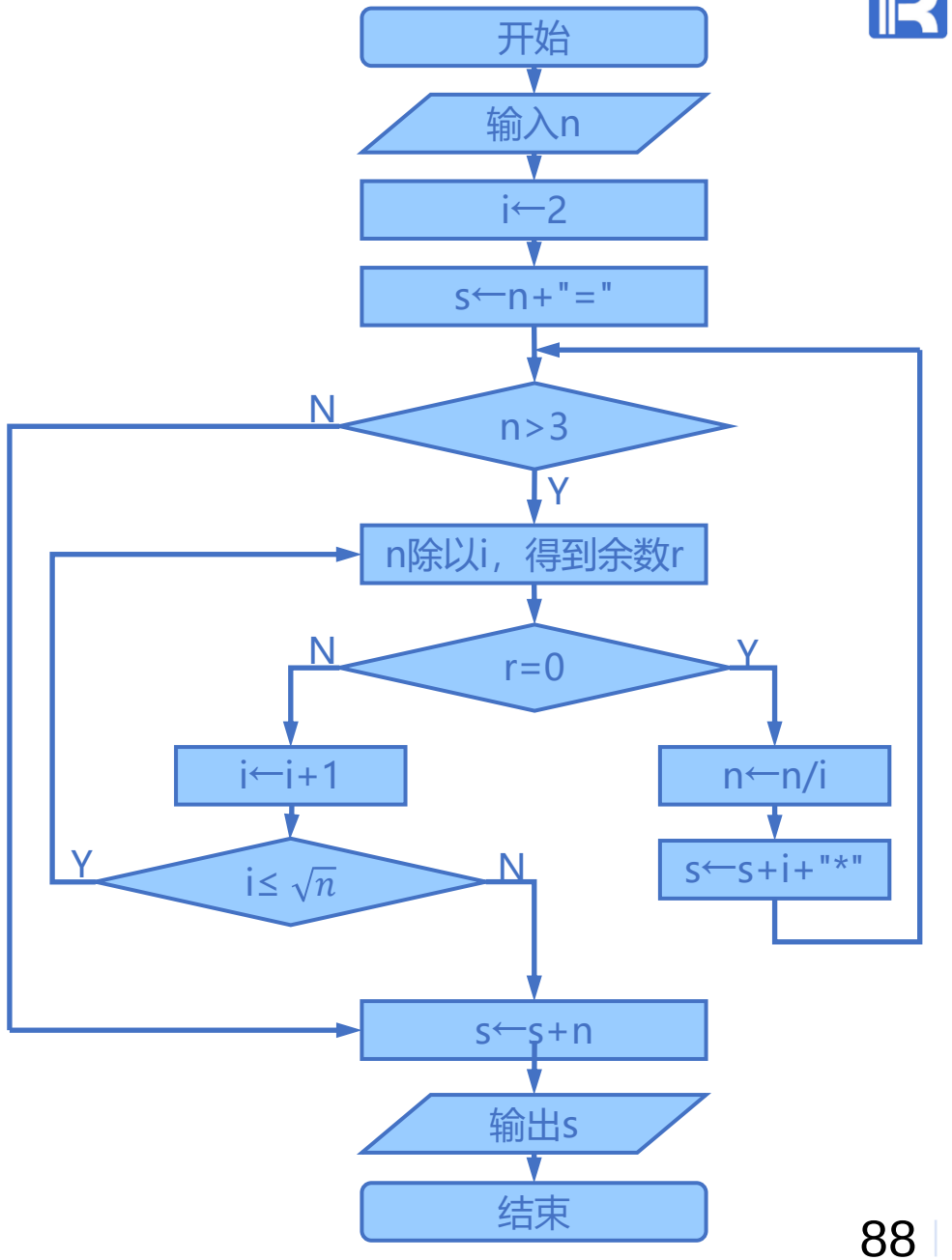
Step4: 计算n除以i的余数r

Step5: 若 r=0，说明n能够被i整除，则n←n/i，s←s+i+"*"，返回到Step3，继续执行。

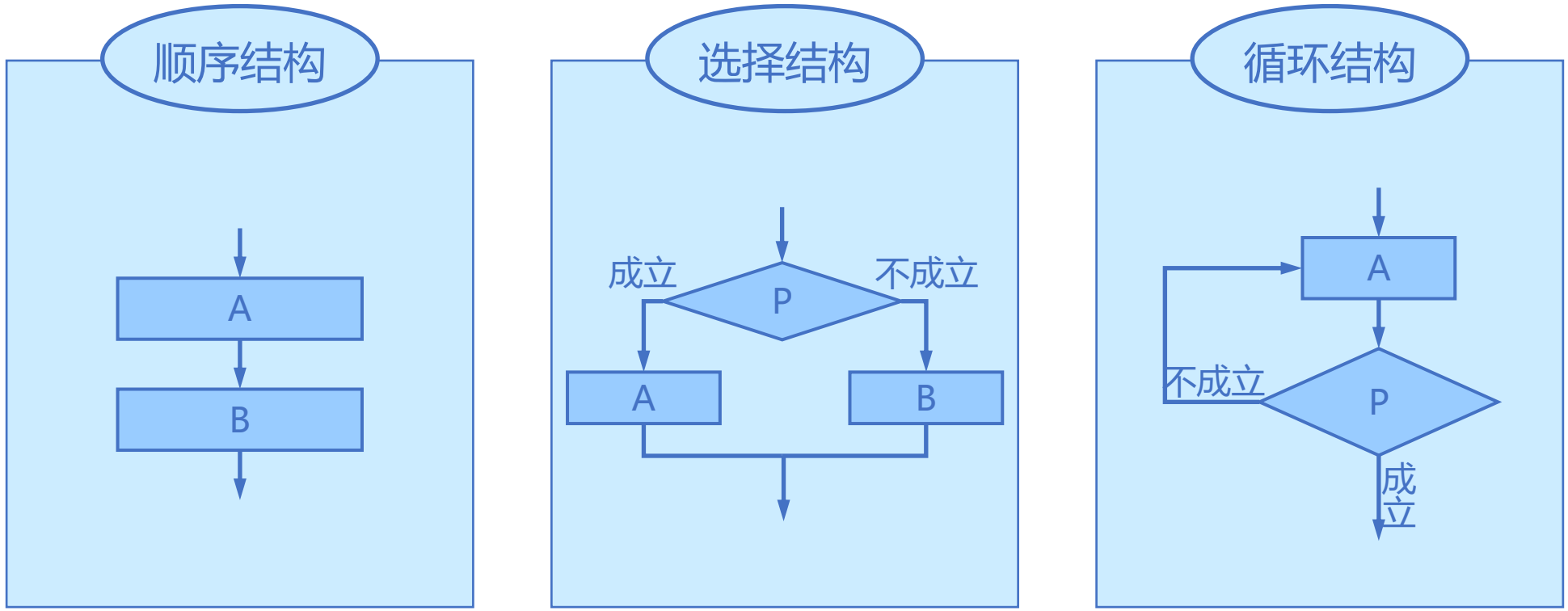
Step6: i←i+1

Step7: 若 i≤$\sqrt{n}$，返回到Step4继续执行。

Step8: s←s+n，输出s并结束算法。



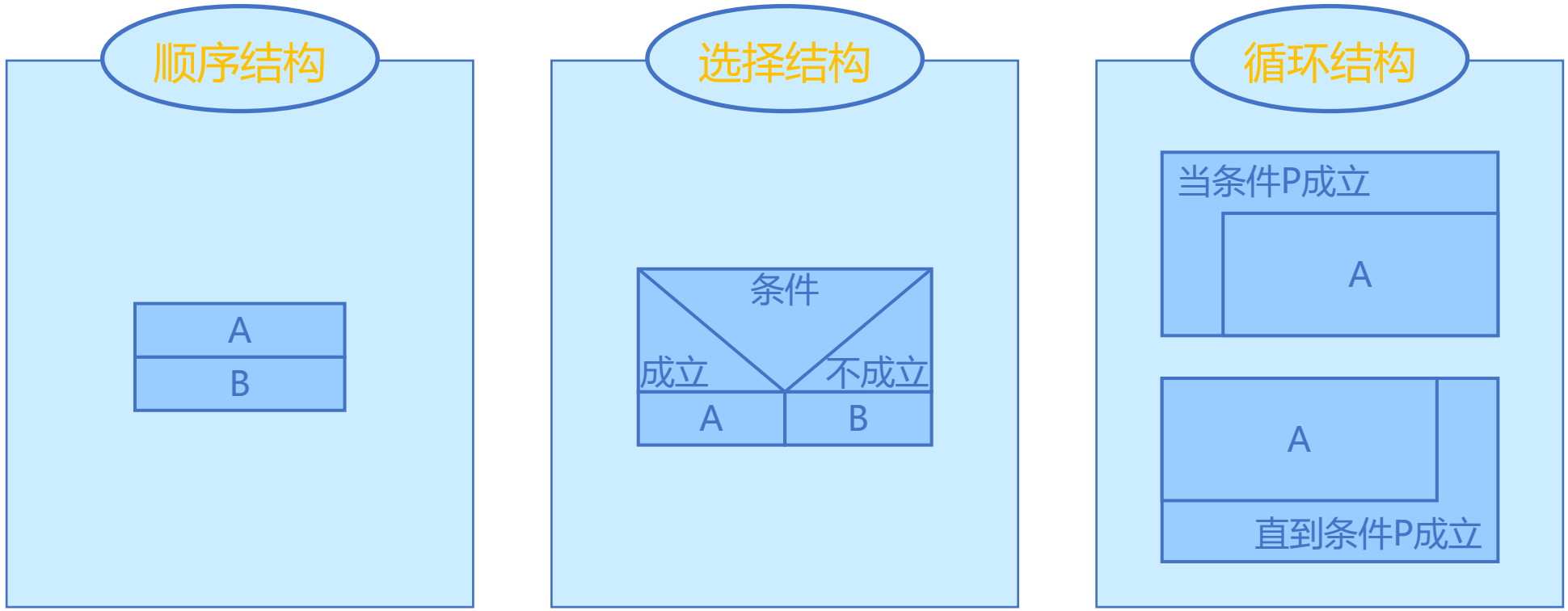
流程图的三种基本结构



三种结构的共同特点

- 只有**一个入口**。
- 只有**一个出口**。
- 结构内的每一个部分都有机会被执行到。
- 结构内不存在“死循环”(不会结束的循环)

如何用N-S流程图表示算法



【例2.11】将例2.1的算法用N-S流程图表示。
计算5×5×5×5×5的结果。

算法步骤

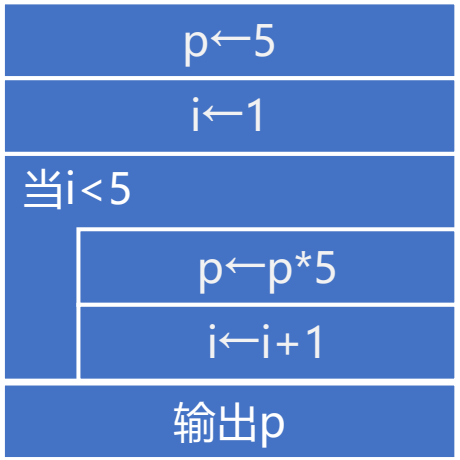
Step1: p←5

Step2: i←1

Step3: p←p*5

Step4: i←i+1

Step5: 若 i≥5，输出p结束；否则返回Step3继续执行。



【例2.12】将例2.4的算法用N-S流程图表示。

请列出2019年之后的连续30个闰年。

算法步骤

Step1: year←2019

Step2: num←0

Step3: 若year不能被4整除，则这一年不是闰年，转到Step7继续执行，检查下一年的情况。

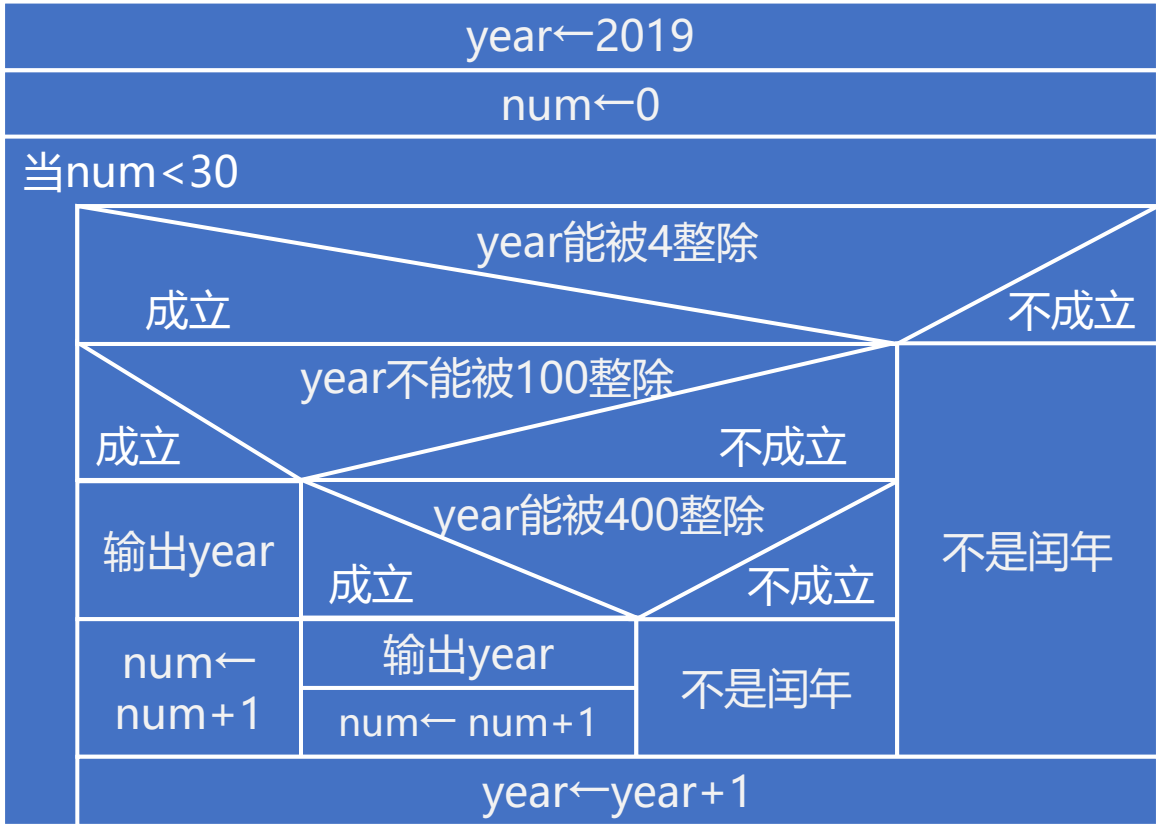
Step4: 若year不能被100整除，则这一年是闰年，转到Step6继续执行；否则继续判断。

Step5: 若year能被400整除，则这一年是闰年，继续执行；否则说明不是闰年，跳转到Step7。

Step6: 找到一个闰年，输出year，num←num+1

Step7: year←year+1

Step8: 若 num<30，转到Step3继续执行，判断下一年；否则寻找完成，算法结束。



什么是伪代码

- 非正式的，结构介于英语(或中文等自然语言）和计算机语言之间的，用来描述算法的语言。
- 每一行或几行伪代码表示一个基本操作。
- 从上到下地书写，并用缩进表示算法中的分支结构。
- 结构清晰，代码简单，可读性好；但可能缺乏严格性

【例2.13】将例2.1的算法用伪代码表示。
计算5×5×5×5×5的结果。

算法步骤

Step1: p←5

Step2: i←1

Step3: p←p*5

Step4: i←i+1

Step5: 若 i≥5，输出p结束；否则返回Step3继续执行。

伪代码

```
begin      (算法开始)
   p←5
   i←1
   while i<5
   {
       p←p*5
       i←i+1
   }
   print p
end        (算法结束)
```

【例2.14】将例2.1的算法用SPL表示。

计算5×5×5×5×5的结果。

算法步骤

Step1: p←5

Step2: i←1

Step3: p←p*5

Step4: i←i+1

Step5: 若 i≥5，输出p结束；否则返回Step3继续执行。

【例2.14】将例2.1的算法用SPL表示。

计算5×5×5×5×5的结果。

算法步骤

Step1: p←5

Step2: i←1

Step3: p←p*5

Step4: i←i+1

Step5: 若 i≥5，输出p结束；否则返回Step3继续执行。

	A	B
1	5	1
2	for B1<5	>A1=A1*5
3		>B1=B1+1
4	=A1	

SPL语法

执行顺序：由上至下，由左至右

for循环 ：当条件满足时，重复执行代码块中的代码

代码块 ：A2为for循环主格，它下面的各行中，如果主格所在列及左侧的所有单元格均为空格，则这些行及主格所在行一起构成主格的代码块，如图中的红框部分

有关SPL的相关使用，将在第4章、第5章中详细讲解，这里仅需简单了解。

【例2.15】将例2.3的算法用SPL表示。

计算：$\pi \approx 4-\frac{4}{3}+\frac{4}{5}-\frac{4}{7}+\frac{4}{9}-\frac{4}{11}+\cdots+\frac{4}{197}-\frac{4}{199}$

算法步骤

Step1: res←4

Step2: sign←-1

Step3: deno←3

Step4: term←sign*(4/deno)

Step5: res←res+term

Step6: sign←sign*(-1)

Step7: deno←deno+2

Step8: 若 deno≤199，返回到Step4，继续执行；否则结束并输出res。

	A	B	C	D
1	4	/A1: res	-1	/C1: sign
2	3	/A2: deno		
3	for A2<=199	=C1*(4.0/A2)	/B3: term	
4		>A1=A1+B3		
5		>C1=C1*(-1)		
6		>A2=A2+2		
7	>output(A1)	/输出π		

【例2.16】将例2.4的算法用SPL表示。

请列出2019年之后的连续30个闰年。

算法步骤

Step1: year←2019

Step2: num←0

Step3: 若year不能被4整除，则这一年不是闰年，转到Step7继续执行，检查下一年的情况。

Step4: 若year不能被100整除，则这一年是闰年，转到Step6继续执行；否则继续判断。

Step5: 若year能被400整除，则这一年是闰年，继续执行；否则说明不是闰年，跳转到Step7。

Step6: 找到一个闰年，输出year，num←num+1

Step7: year←year+1

Step8: 若 num<30，转到Step3继续执行，判断下一年；否则寻找完成，算法结束。

	A	B	C	D
1	2019	/A1: year	0	/C1: num
2	for C1<30	if A1%4 == 0		
3			if A1%100 != 0	>output(A1)
4				>C1=C1+1
5			else if A1%400 == 0	>output(A1)
6				>C1=C1+1
7		>A1=A1+1		

2.5 结构化程序设计方法

2.5 结构化程序设计方法

结构化程序设计

- 分析需求时，**自顶向下，逐步细化**。
- 面向结构，根据功能**划分模块**。
- **结构化编码**，将细化的算法用计算机语言表示。

2.6 习题

2.6 习题

【习题2.1】从日常生活中分别找出一个数值运算问题和一个非数值运算问题，描述它们的算法。

【习题2.2】根据流程图的三种分支结构例子，设计一个不同的选择流程图和一个不同的循环流程图。

【习题2.3】将下面问题的算法分别用流程图、NS流程图和伪算法表示：
计算：1+3+5+7+……+197+199。

【习题2.4】将下面问题的算法分别用流程图、NS流程图和伪算法表示：
输入圆的周长，求它的周长和面积，π取3.14。

【习题2.5】将下面问题的算法分别用流程图、NS流程图和伪算法表示：
输入某个学生的4科考试成绩，计算他的总分。

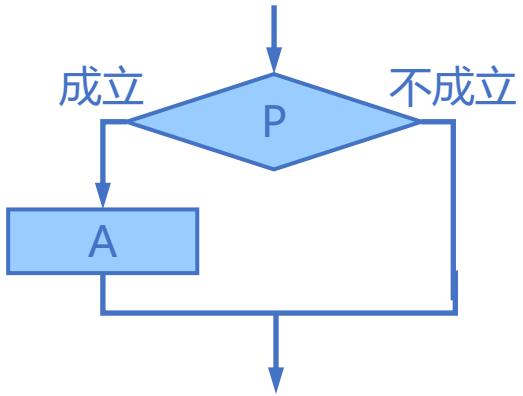
【习题2.6】将下面问题的算法分别用流程图、NS流程图和伪算法表示：
输出斐波那契数列的前30个数：1，1，2，3，5，8，13,…。

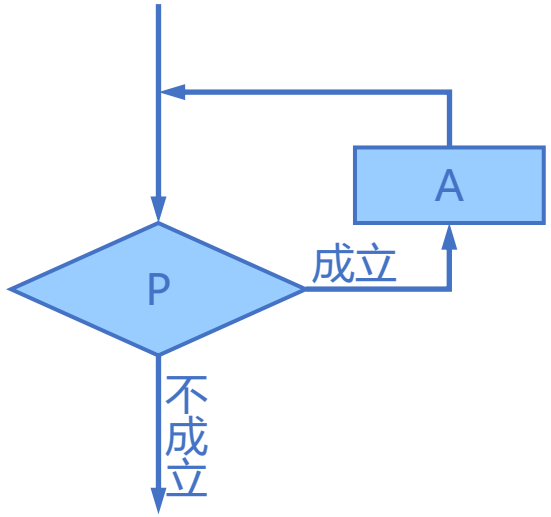
【习题2.7】将下面问题的算法分别用流程图、NS流程图和伪算法表示：
找出所有的3位水仙花数：它所有位上数字的立方和，等于它本身。

【习题2.8】将下面问题的算法分别用流程图、NS流程图和伪算法表示：
鸡翁一值钱伍；鸡母一值钱三；鸡雏三值钱一。百钱买鸡百只，问鸡翁、母、雏各几何？

2.6 习题

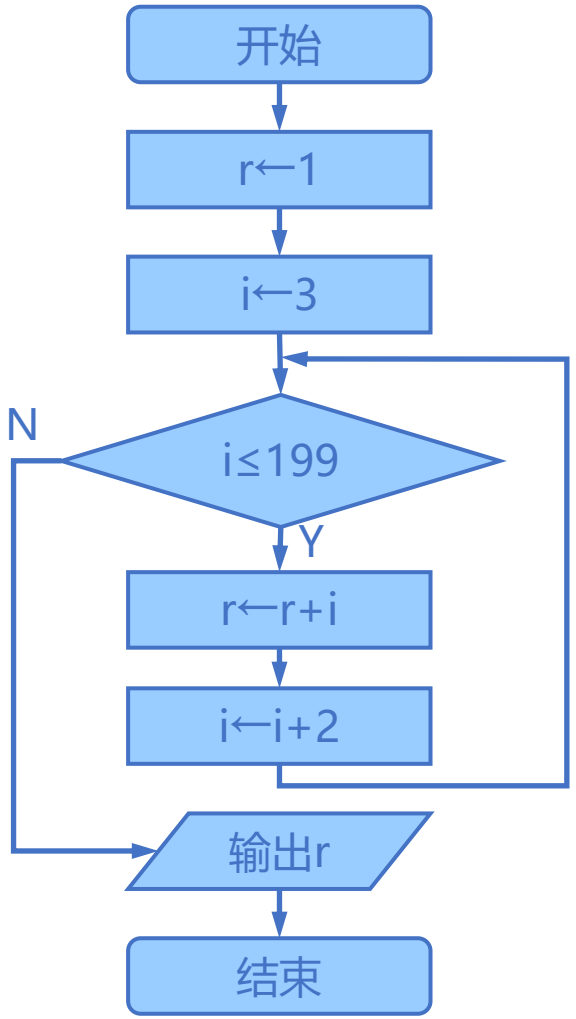
【习题2.2】根据流程图三种分支结构的例子，设计与之不同的一个选择流程图和一个循环流程图。





2.6 习题

【习题2.3】将下面问题的算法分别用流程图、NS流程图和伪算法表示：
计算：1+3+5+7+……+197+199。



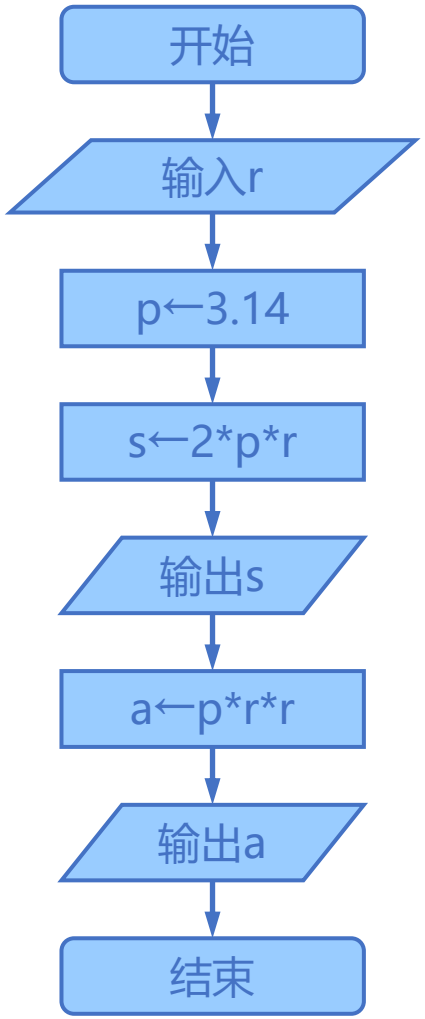
伪代码

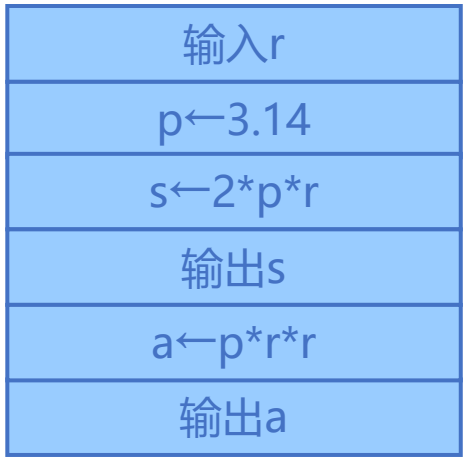
```
begin      (算法开始)
  r←1
  i←3
  while i≤199
  {
      r←r+i
      i←i+2
  }
  print r
end        (算法结束)
```

2.6 习题

【习题2.4】将下面问题的算法分别用流程图、NS流程图和伪算法表示：

输入圆的周长，求它的周长和面积，π取3.14。



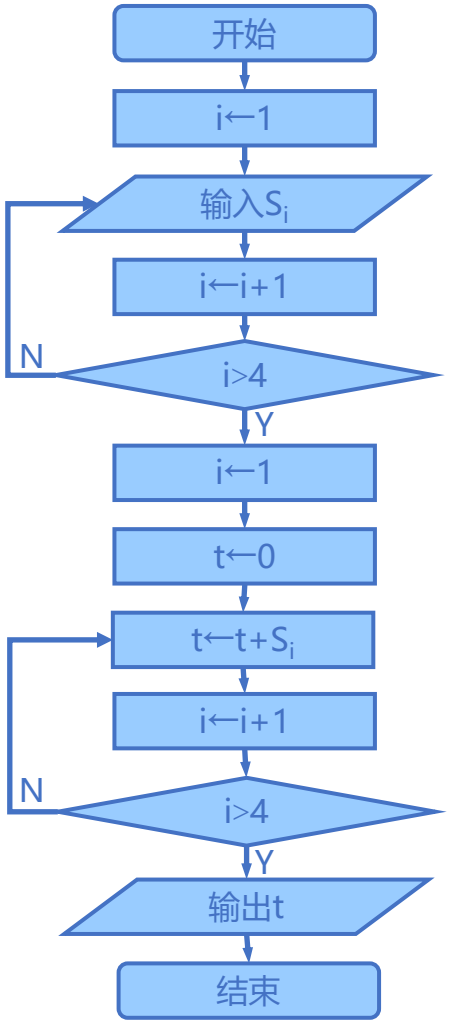


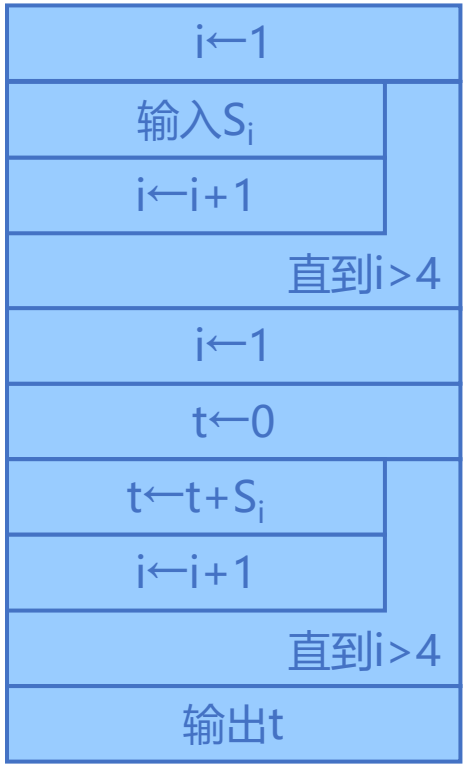
伪代码

```
begin      (算法开始)
   read r
   p←3.14
   s←2*p*r
   print s
   a←p*r*r
   print a
end        (算法结束)
```

2.6 习题

【习题2.5】将下面问题的算法分别用流程图、NS流程图和伪算法表示：
输入某个学生的4科考试成绩，计算他的总分。



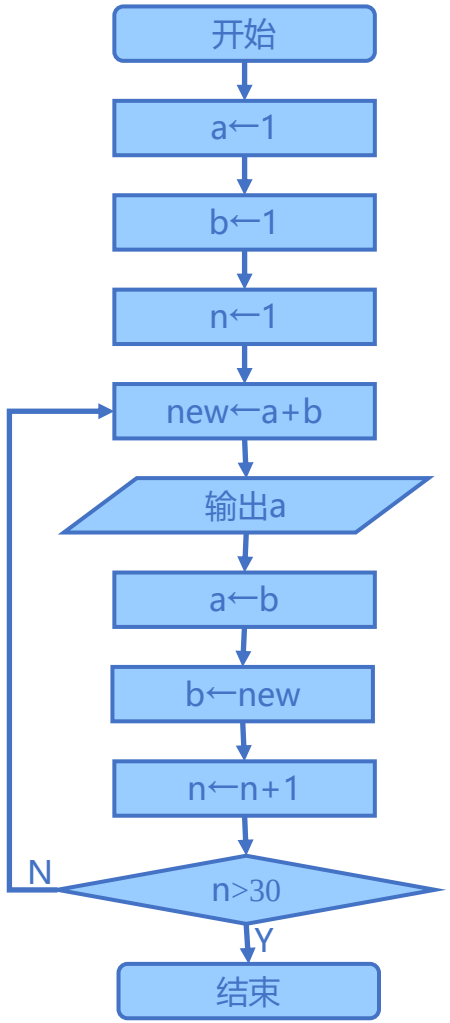


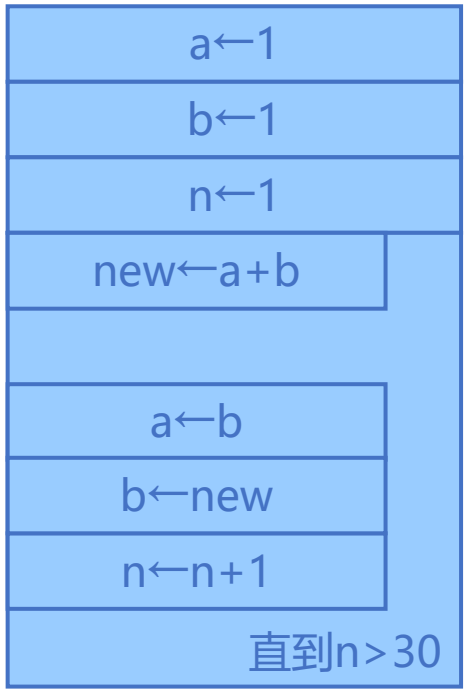
伪代码

```
begin        (算法开始)
   i←1
   do {
      read Si
      i←i+1
   }
   while i≤4
   i←1
   t←0
   do {
      t←t+Si
      i←i+1
   }
   while i≤4
   print t
end          (算法结束)
```

2.6 习题

【习题2.6】将下面问题的算法分别用流程图、NS流程图和伪算法表示：
输出斐波那契数列的前30个数：1, 1, 2, 3, 5, 8, 13,…。



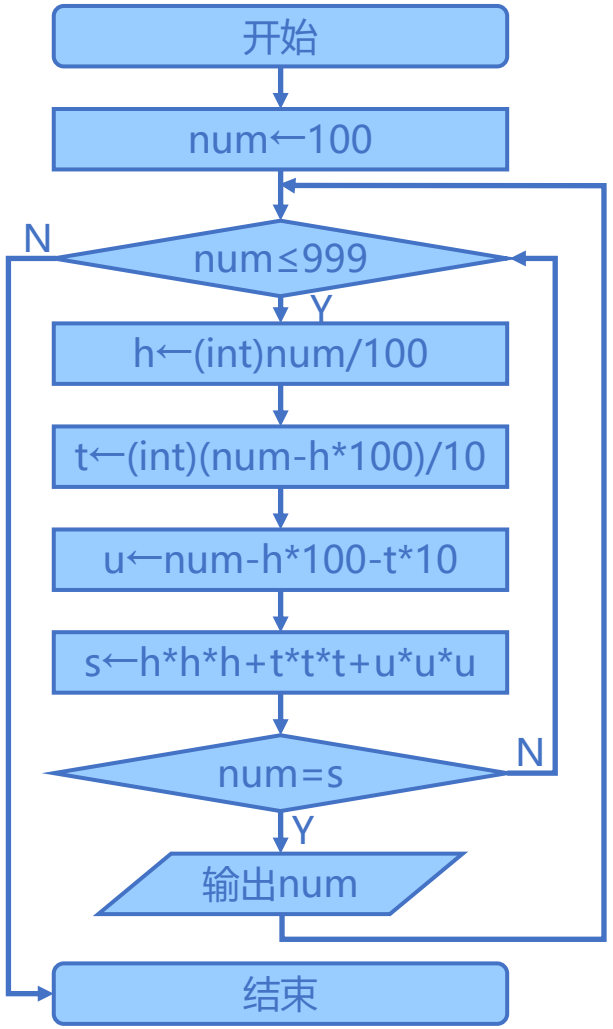


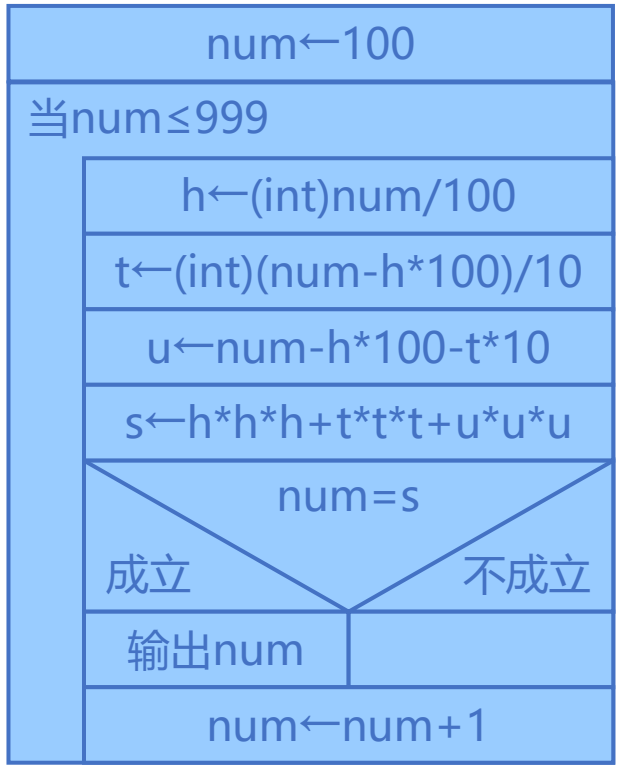
伪代码

```
begin        (算法开始)
  a←1
  b←1
  n←1
  do {
      new←a+b
      print a
      a←b
      b←new
      n←n+1
  }
  while n≤30
end          (算法结束)
```

2.6 习题

【习题2.7】将下面问题的算法分别用流程图、NS流程图和伪算法表示：
找出所有的3位水仙花数：它所有位上数字的立方和，等于它本身。





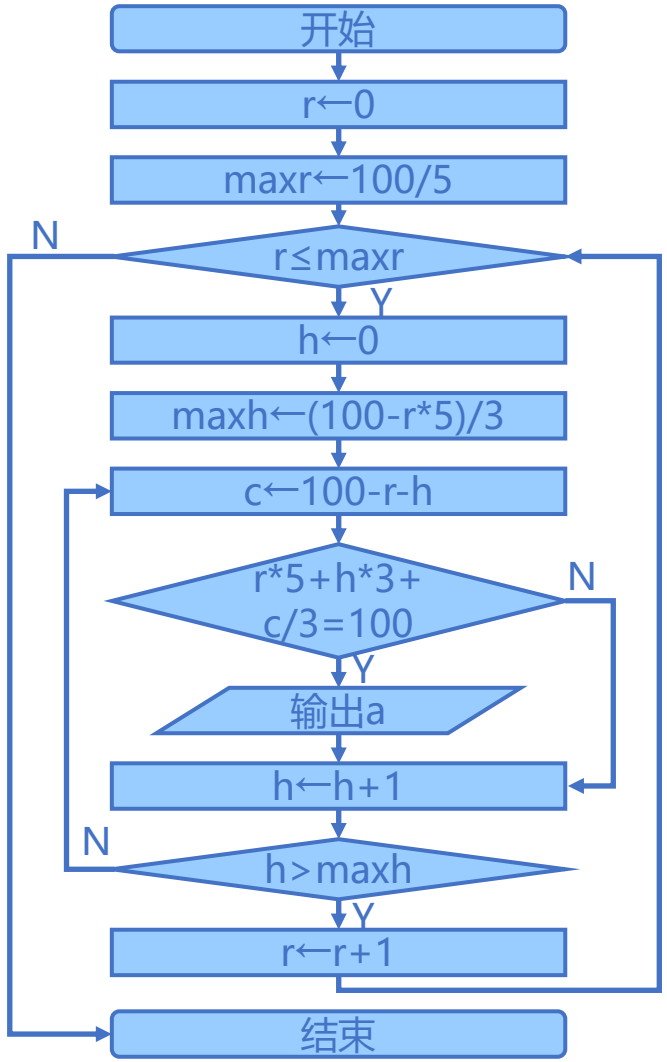
伪代码

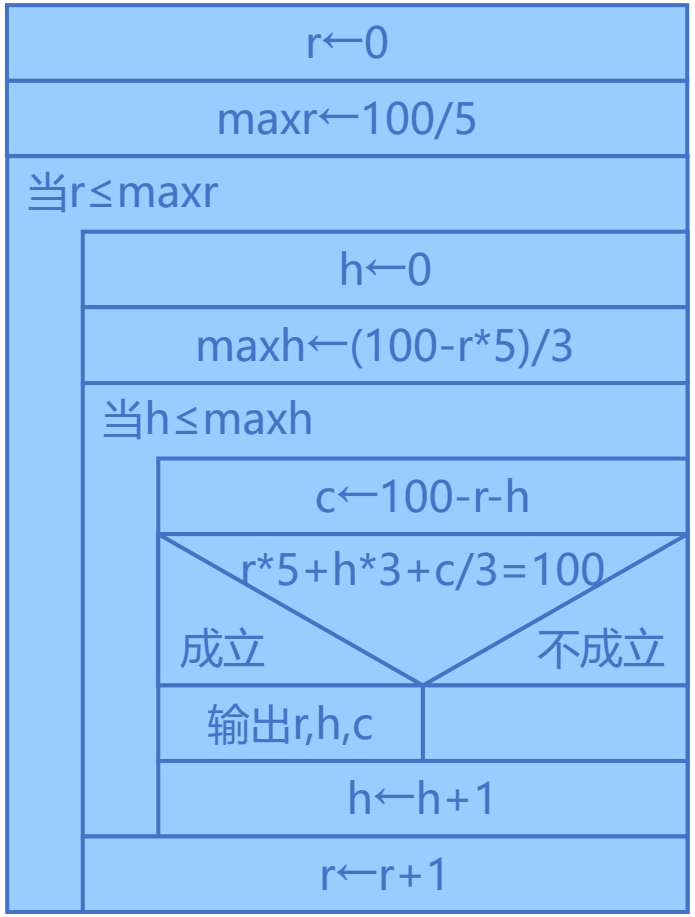
```
begin       (算法开始)
  num←100
  while num≤999
  {
      h←(int)num/100
      t←(int)(num-h*100)/10
      u←num-h*100-t*10
      if num=h*h*h+t*t*t+u*u*u
      {
          print num
      }
      num←num+1
  }
end         (算法结束)
```

2.6 习题

【习题2.8】将下面问题的算法分别用流程图、NS流程图和伪算法表示：

鸡翁一值钱伍；鸡母一值钱三；鸡雏三值钱一。百钱买鸡百只，问鸡翁、母、雏各几何？





伪代码

```
begin       (算法开始)
  r←0
  maxr←100/5
  while r≤maxr
  {
     h←0
     maxh←(100-r*5)/3
     while h≤maxh
     {
        c←100-r-h
        if r*5+h*3+c/3=100
        {
           print r,h,c
        }
        h←h+1
     }
     r←r+1
  }
end         (算法结束)
```

第3章 顺序结构程序设计

3.1 顺序程序设计举例

3.2 数据的表现形式及其运算

- 常量和变量
- 数据类型
- 数据类型的识别

3.3 运算符和表达式

- SPL中常用的运算符
- 基本的算术运算符
- 运算符的优先级
- 多类型数据混合运算
- 强制类型转换运算符

3.4 SPL语句

3.5 数据的输入输出

- 输入输出问题举例
- 在SPL中输入输出数据

3.6 习题

3.1 顺序程序设计举例

3.1 顺序程序设计举例

【例3.1】某商品月初的存量是100件，1日进货20件后，售出了全部商品的2/3；2日又进货库存余量的3倍，并售出了135件。2日营业结束后，商品库存余量是多少?

解题思路

根据每天的进货量和销售数量，依次计算每天的商品库存。

N-S流程图

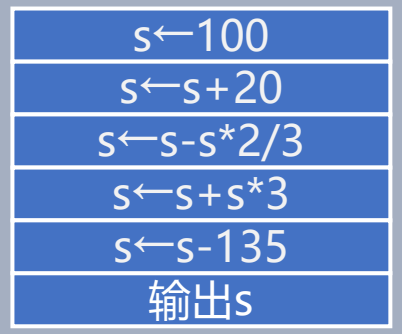
3.1 顺序程序设计举例

【例3.1】某商品月初的存量是100件，1日进货20件后，售出了全部商品的2/3；2日又进货库存余量的3倍，并售出了135件。2日营业结束后，商品库存余量是多少?

解题思路

根据每天的进货量和销售数量，依次计算每天的商品库存。

N-S流程图



	A	B
1	>s=100	/用变量s记录商品库存量，指定s的初值为100
2	>s=s+20	/计算1日进货后的库存
3	>s=s-s*2/3	/计算1日销售后的剩余库存
4	>s=s+s*3	/计算2日进货后的库存
5	>s=s-135	/计算2日销售后的剩余库存
6	=s	/获取s的最终结果

3.1 顺序程序设计举例

【例3.1】某商品月初的存量是100件，1日进货20件后，售出了全部商品的2/3；2日又进货库存余量的3倍，并售出了135件。2日营业结束后，商品库存余量是多少?

解题思路

根据每天的进货量和销售数量，依次计算每天的商品库存。

N-S流程图

s←100
s←s+20
s←s-s*2/3
s←s+s*3
s←s-135
输出s

	A	B
1	>s=100	/用变量s记录商品库存量，指定s的初值为100
2	>s=s+20	/计算1日进货后的库存
3	>s=s-s*2/3	/计算1日销售后的剩余库存
4	>s=s+s*3	/计算2日进货后的库存
5	>s=s-135	/计算2日销售后的剩余库存
6	=s	/获取s的最终结果

Value
25.0

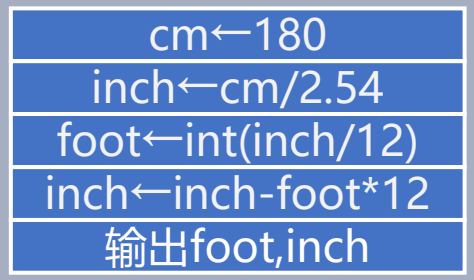
3.1 顺序程序设计举例

【例3.2】英国惯用的身高是和公制不同的，如160cm约相当于5'3。那么如果某人的身高是180cm，相当于几英尺几英寸呢？（1英寸=2.54厘米，计算时保留1位小数）

解题思路

先将厘米数转换为英寸数，再计算合多少英尺多少英寸。

N-S流程图



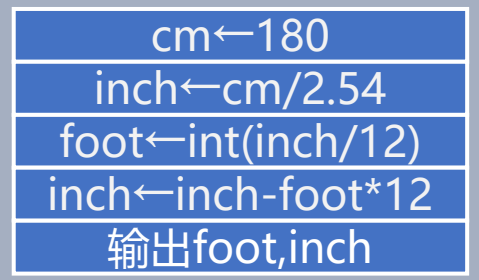
3.1 顺序程序设计举例

【例3.2】英国惯用的身高是和公制不同的，如160cm约相当于5'3。那么如果某人的身高是180cm，相当于几英尺几英寸呢？（1英寸=2.54厘米，计算时保留1位小数）

解题思路

先将厘米数转换为英寸数，再计算合多少英尺多少英寸。

N-S流程图



	A	B
1	>cm=180	/用变量cm记录身高厘米数
2	>inch=cm/2.54	/计算折合的英寸数
3	>foot=inch\12	/计算英尺部分数值
4	>inch=round(inch-foot*12,1	/计算英寸部分数值，保留1位小数
5	=foot/"'"/inch	/将英尺与英寸数合并

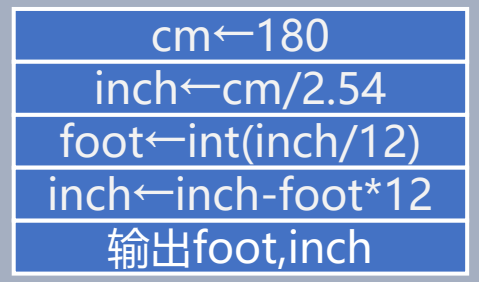
3.1 顺序程序设计举例

【例3.2】英国惯用的身高是和公制不同的，如160cm约相当于5'3。那么如果某人的身高是180cm，相当于几英尺几英寸呢？（1英寸=2.54厘米，计算时保留1位小数）

解题思路

先将厘米数转换为英寸数，再计算合多少英尺多少英寸。

N-S流程图



	A	B
1	>cm=180	/用变量cm记录身高厘米数
2	>inch=cm/2.54	/计算折合的英寸数
3	>foot=inch\12	/计算英尺部分数值
4	>inch=round(inch-foot*12,1	/计算英寸部分数值，保留1位小数
5	=foot/"'"/inch	/将英尺与英寸数合并

Value
5'10.9

3.2 数据的表现形式及其运算

常量

也叫常数

程序运行中保持不变的数据

变量

也叫变数

程序运行中可以变化的数据

标识符

- **标识符**就是一个对象的名称，可以用于标识**常量、变量**等。
- 标准的标识符只能由**字母、数字**或**下划线**_构成，如果需要也可使用汉字，但需要避免包含可能产生歧义的各种符号，如加号+逗号,小数点.等。
- 有些程序语言的标识符会**区分大小写字母**，即A和a是不同的。标识符的名字最好能表示它的作用与含义。
- 标识符应该**避免使用关键字和数值**等。

SPL中的部分关键字

abs	acos	acosh	age	and	arguments	asc	asin	asinh	atan
	atanh	base64	between	bits	break	call	callx	canvas	case
	ceil	channel	char	clear	clipboard	cmp	combin	concat	connect
	cos	cosh	create	cum	cursor	date	datetime	day	days
	deq	directory	elapse	else	end	env	eval	exp	fact
	false	file	filename	fill	floor	for	fork	format	func
	gcd	goto	hash	hosts	hour	httpfile	if	ifa	ifdate
	ifnumber	ifr	ifstring	ift	iftime	ifv	inf	interval	invoke
	isalpha	isdigit	islower	isupper	iterate	join	json	k	lcm
	left	len	lg	like	ln	lock	lower	md5	mid
	millisecond	minute	month	movefile	next	not	now	null	num
	number	or	ord	output	pad	parse	pdate	periods	permut
	pgall	pgcell	pgno	pi	pos	power	product	rand	rands
	range	rank	ranki	reduce	register	replace	return	rgb	right
	round	row	second	seq	shift	sign	sin	sinh	sizeof
	sleep	sqrt	string	substr	sync	system	tan	tanh	time
	to	trim	true	upper	urlencode	vdbase	workday	workdays	xjoin
	xjoinx	xml	xor	year	z	zone			

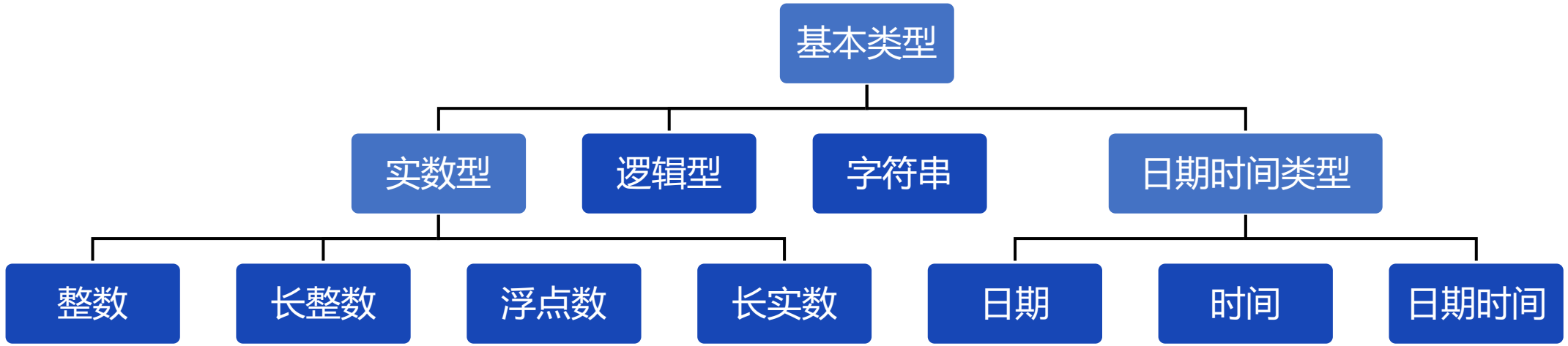
单元格变量

- 名称由单元格所在位置决定，由**列号字母和行号数字**构成，如B1表示第1行第2列。
- 单元格中可以直接输入数据作为单元格变量的初始值。
- 不能另行命名，直接使用单元格名访问。
- 当单元格变量被赋值时，相当于单元格本身被赋值。

	A	B	C
1		12.345	
2			
3			

单元格变量是SPL中特有的，一般程序设计语言写成文本，没有单元格变量

基本数据类型



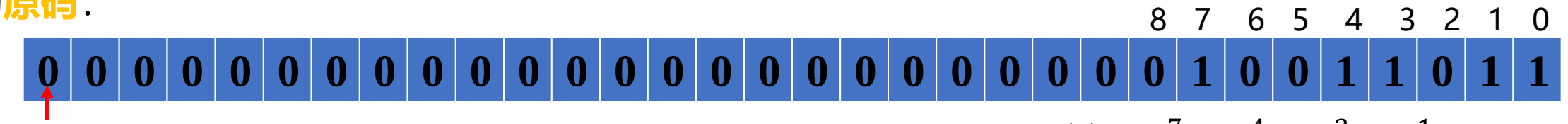
实数型中各种类型的分别，将在本节后面详细讲解。

整数在计算机中的存储方式

SPL中的整数是32位的带符号整数，存储为32个bit。

符号位，正数为0，负数为1

155的**原码**：



正数的原码符号位为0

$$155 = 2^7 + 2^4 + 2^3 + 2^1 + 1$$

155的**反码**：正整数的反码和原码相同

155的**补码**：正整数的补码也等于原码

整数在计算机中的存储方式

-155的**原码**：

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1

负数的原码符号位为1

$155 = 2^7 + 2^4 + 2^3 + 2^1 + 1$

-155的**反码**：负整数的反码除符号位以外都取反，0变为1,1变为0.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	0	0

-155的**补码**：负整数的补码等于反码+1

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	0	1

整数的**补码是计算机中整数的真正存储方式**，原码便于理解但不便于计算。

整数在计算机中的存储方式

最大的正整数：$2^{31}-1$ =2147483647

0:

-1:

最小的负整数：-2^{31}= −2147483648

这些内容有个印象即可，知道计算机中的二进制表示有些特点，细节不理解并不影响后续学习

整数和长整数类型

整数和长整数的对比

数据类型	位数	字节数	取值范围
整数	32位	4	-2147483648~2147483647
长整数	64位	8	-2^{63}~$2^{63}-1$

► 使用整数或者长整数时，需要注意数值本身或者计算时产生的结果是否超出了**取值范围**。

浮点数类型

浮点数

- SPL中的浮点数是**64位**的**双精度浮点数**，存储为64位(8个字节)。
- 浮点数的取值范围是：0以及
  正数： 2.2250738585072014E-308 ~ 1.7976931348623157E+308
  负数：-1.7976931348623157E+308 ~ -2.2250738585072014E-308
- 浮点数是用二进制存储，有一些数值是无法精确表示的，所以在浮点数的运算中存在误差，如0.05+0.01得到的结果是0.06000000000000005

长实数类型

长实数

- 长实数可以**无误差**地存储任何实数。
- 长实数需要占用更多的内存，执行计算时**效率会较低**。
- 无误差的长实数，通常应由**字符串**转换获得；也可由其它**数值**转换获得。
- 如果是将浮点数转换为长实数，由于浮点数本身可能存在误差，因此得到的长实数仍然**可能不准确**。

字符串类型

字符串

- 字符串类型可以是任意长度的一串字符，也可以是空格字符串甚至空字符串。
- 组成字符串中的每一个字符，都可以转换为一个对应的非负整数。
- 共有128个基本字符，在ASCII码表中由整数0~127分别对应。
- 与数值不同，字符串参与的是查找字符、替换字符、截取子字符串之类操作。

有关字符串的使用，将在第6章中详细讲解，这里只需初步了解。

逻辑类型

逻辑类型（**布尔型**）

- 只有**true**(真)和**false**(假)两个值。
- 逻辑类型数据用来参与逻辑运算，或者表示逻辑运算的结果。

日期及时间类型

日期/时间类型

- 包括**日期型，时间型**和**日期时间型**3种类型。分别用来表示日期、时间、或者同时表示日期和时间。
- 日期及时间类型可与**特定格式**的字符串相互转换。同一个日期可能转换为不同格式的字符串，如2019年3月14日，可能表示为"2019-3-14", "3/14/19"或者"14/3/2019"。
- 日期及时间类型数据，会参与一些特殊的计算，而且计算往往有着多种模式，如计算差值时，可能需要获得两个日期相差多少年来获得年龄，也可能需要获得相差多少天或相差多少周等结果。
- 使用日期及时间数据时，可以使用默认格式，也可以根据情况定义格式。

常量的类型

SPL对常量类型的判定

- 不带小数点的数值是**整数**类型，其中超过32位整数取值范围的为**长整数**，长整数也可以用整数后加字母**L**表示。
- 带小数点的数值或者超过长整数取值范围的整数，会被判定为**浮点数**，浮点数也可以用数值后加字母**f**或**d**表示。如果数值超出了浮点数的取值范围会报错，长实数无法直接作为常量定义。
- 整数、长整数和浮点数前可以带**负号-**或者**正号+**。
- 符合默认日期/时间**格式**的字串会被判定为日期及时间类型数据。
- 只有小写的**true**和**false**会被判定为布尔类型。
- 其它字串将被判定为字符串型数据，但单元格中的字符串常量应**避免使用关键字**。字符串常量也可以在字串前加**单引号'**表示。

字串	类型
123	整数
-12345678901	长整数
+3.1416	浮点数
false	布尔数
2019-03-25	时间
The brown fox.	字符串

单元格值的显示

单元格值的显示区别

- **整数**显示为蓝色，靠右对齐。
- **长整数**显示为**黑色**，靠右对齐。
- **浮点数**显示为**深紫色**，靠右对齐。
- **长实数**显示为**红色**，靠右对齐。
- **布尔数**显示为**黑色**，靠左对齐。
- **日期时间类型数值**显示为**黑色**，靠左对齐。
- **字符串**显示为**黑色**并带有**蓝色**下划线，靠左对齐。

类型	单元格值的显示
整数	Value 123
长整数	Value -12345678901
浮点数	Value 3.1416
长实数	Value 0.06
布尔数	Value false
日期时间	Value 2019-03-25
字符串	Value The brown fox.

3.3 运算符和表达式

SPL中常用的运算符

分类	运算符
算术运算符	+ - * / % \
字符串连接符	+ /
赋值运算符	= ?=
关系运算符	> < == >= <= !=
逻辑运算符	&& || !
逗号运算符	$(x_1, x_2, x_3, \ldots x_k)$
类型转换运算符	int() long() float() decimal() date() time() datetime() string() ""
成员运算符	. .()
序列运算符	() | \ ^ & %

单目运算符和双目运算符

单目运算符

- 单目运算符是指只需要**一个操作数**的运算符，也叫一元运算符。
- 单目运算符包括：

运算符	含义
+ -	正号、负号运算符
!	逻辑非运算符
int() long() float() decimal() date() time() datetime() string() ""	类型转换运算符

双目运算符

- 双目运算符是指所需**操作数为两个**的运算符，也叫二元运算符。
- 双目运算符包括：

运算符	含义
+ - * / % \	加法、减法、乘法、除法、求余、整除运算符
+ /	字符串连接运算符
= ?=	赋值运算符
> < == >= <= !=	关系运算符
&& ||	逻辑与、或运算符
() | \ ^ & %	序列运算符

运算符	含义	举例	结果
+	正号运算符	+a	a的值
+	加法运算符	a+b	a加b的和
-	负号运算符	-a	a的负值
-	减法运算符	a-b	a减b的差
*	乘法运算符	a*b	a乘以b的积
/	除法运算符	a/b	a除以b的商
%	求余运算符	a%b	a除以b得到的余数
\	整除运算符	a\b	a除以b商的整数部分

算术表达式与运算的优先级

- 用算数运算符和括号将操作数连接起来的，符合SPL语法规则的，能够实现计算要求的式子称为**SPL算术表达式**。
- 算术表达式中，不同的运算符有着不同的**优先级**，在计算时会先计算优先级较高的，先计算乘、除、整除、求余等操作符，再计算加减法。
- 在计算普通的算术表达式时，将按照从左至右的顺序计算，又称为“**左结合性**”；但在处理赋值运算符时，将先计算右侧的部分，这称为“**右结合性**”。
- 在SPL常用的运算符中，括号优先级是最高的第1级，其次是第2级成员运算符，正负号是第3级，第4级是非运算符，第5级是其它算术运算符，第6级是关系运算符，第7级是其它逻辑运算符，第8级是赋值运算符，逗号操作符的优先级是最低的第9级。

不同类型实数的**算术运算**

- 实数的四种类型中，**级别排序**是：
  长实数>浮点数>长整数>整数，通常的算术运算结果，其类型是参与计算的数据中级别较高的。如长整数减整数的差为长整数，长实数与整数的和为长实数等。

- 求余运算符只允许在**整数**或者**长整数**间使用。

- 如果参与除法计算的数据中没有长实数，那么结果为**浮点数**，即使整数与整数相除，结果也是浮点数。

- 执行整除运算时，只有两个整数整除的结果才是**整数**，其它情况结果都是**长整数。**

字符串的连接运算

- 在SPL中，字符串连接符有两个：+和/。
- 字符串连接符+要求两个操作数必须都是字符串，其它类型的操作数必须用类型转换符string()或者format()，char()等函数先行转换为字符串。
- 字符串连接符/要求两个操作数至少有一个是字符串，另一个操作数如果不是字符串将会自动转换。
- 因为对操作数的要求不同，因此字符串连接符和算术运算符中的加法运算符和除法运算符并不会混淆。

类型名(表达式)

类型	强制类型转换符
整数	int()
长整数	long()
浮点数	float()
长实数	decimal()
布尔数	bool()
日期时间	date() time() datetime()
字符串	string()

3.4 SPL语句

3.4 SPL语句

控制语句

if … else …	条件语句
goto	跳转语句
for …	循环语句
fork … reduce …	多线程循环语句
next	继续循环语句
break	结束循环语句
return	结束当前子程序，并返回结果
end	主动异常结束，并返回异常信息

表达式语句

- 表达式语句有两种：赋值语句和执行语句，对应着计算格与执行格。
- 赋值语句以=开头，后面跟随一个表达式，执行后将把表达式的结果赋值给单元格。如=3
- 执行语句以>开头，后面跟随一个表达式，执行的表达式不一定会有结果，有结果时也不会赋值给单元格。如>A1=A1+4

【例3.3】已知三角形的三边长分别是5.6,7.8和9，请问三角形的面积是多少？可以使用海伦公式：$S_\Delta = \sqrt{p(p-a)(p-b)(p-c)}$，其中 a, b, c 为三角形三边长，$p = \frac{a+b+c}{2}$。

【例3.3】已知三角形的三边长分别是5.6,7.8和9，请问三角形的面积是多少？可以使用海伦公式：$S_\Delta = \sqrt{p(p-a)(p-b)(p-c)}$，其中 a, b, c 为三角形三边长，$p = \frac{a+b+c}{2}$。

解题思路

用单元格常数获得三角形三边长，使用海伦公式，先计算出p的值，再计算三角形面积。

【例3.3】已知三角形的三边长分别是5.6,7.8和9，请问三角形的面积是多少？可以使用海伦公式：$S_\Delta = \sqrt{p(p-a)(p-b)(p-c)}$，其中 a, b, c 为三角形三边长，$p = \frac{a+b+c}{2}$。

解题思路

用单元格常数获得三角形三边长，使用海伦公式，先计算出p的值，再计算三角形面积。

	A	B	C	D
1	5.6	7.8	9	/输入三角形三边长
2	=(A1+B1+C1)/2			/计算海伦公式中的p
3	=sqrt(A2*(A2-A1)*(A2-B1)*(A2-C1))			/用海伦公式计算三角形面积

Value
21.659769158511356

注意

sqrt()是计算平方根的函数

赋值运算符=

- 等号运算符的作用，是将右侧的数值赋值给左侧的变量，如果左侧没有变量名，将赋值给当前的单元格变量。

如例3.3中的常数格A1，B1和C1也可以用赋值运算符来赋值。

	A	B	C	D
1	=5.6	=7.8	=9	/输入三角形三边长
2	=(A1+B1+C1)/2			/计算海伦公式中的p
3	=sqrt(A2*(A2-A1)*(A2-B1)*(A2-C1))			/用海伦公式计算三角形面积

赋值表达式

- 等号运算符后如果是**表达式**，将先计算右侧的表达式，再将结果赋值给左侧的变量，如果左侧无变量名将赋值给当前单元格。
- 注意赋值表达式将按照“**自右至左**”的顺序计算。

如：

	A	B	C
1	5.6	7.8	9
2	=A1=B1+C1		

赋值表达式

- 等号运算符后如果是**表达式**，将先计算右侧的表达式，再将结果赋值给左侧的变量，如果左侧无变量名将赋值给当前单元格。
- 注意赋值表达式将按照“**自右至左**”的顺序计算。

如：

	A	B	C
1	5.6	7.8	9
2	=A1=B1+C1		

A2中的表达式相当于 =(A1=B1+C1)，即将B1+C1的结果先赋值给A1，再赋值给A2。执行后，A1和A2的格值都会改变，特别是A1中的数值只是它的初值，计算后实际格值已经有了变化：

Value
16.8

Value
16.8

复合赋值运算符

在赋值运算符=之前加上其它数学运算符，可以构成**复合赋值运算符**，如A1+=3，等价于A1=A1+3，要注意前面还要添加=或者>表明表达式是执行计算还是赋值。

各个二元数学运算符都可以和=构成复合赋值运算符。

如：

	A	B	C
1	5.6	7.8	9
2	>A1+=3	>B1/=6	=C1%=4

注意

A2和B2执行后，A1和B1的值会被改变，实际上A1和B1是作为单元格变量使用的。当常数格作为单元格变量使用时，单元格中写入的常数只是初始值，被改变后的值可以选中单元格后在右侧查看。

复合赋值运算符

在赋值运算符=之前加上其它数学运算符，可以构成**复合赋值运算符**，如A1+=3，等价于A1=A1+3，要注意前面还要添加=或者>表明表达式是执行计算还是赋值。

各个二元数学运算符都可以和=构成复合赋值运算符。

如：

	A	B	C
1	5.6	7.8	9
2	>A1+=3	>B1/=6	=C1%=4

执行后，A1，B1，C1和C2的结果如下，由于C2是以等号开头的赋值格，因此C2的格值和C1相同：

Value
8.6

Value
1.3

Value
1

Value
1

调用语句

func	子程序起始语句
return	返回语句
func()	子程序调用语句
call()	跨网格调用语句
callx()	多线程跨网格调用语句
end	主动结束语句

其它语句

clear	清除格值语句
$(db)sql;...	数据库查询语句
cursor()	游标生成语句
for cs	游标循环语句
fork cs	多线程游标循环语句

3.5 数据的输入输出

【例3.4】求二元一次方程组 $\begin{cases} ax+by+m=0 \\ cx+dy+n=0 \end{cases}$ 的解，a,b,m,c,d,n 作为单元格参数输入，如它们的值依次为1,1,-9,2,-2,-10 。

【例3.4】求二元一次方程组 $\begin{cases} ax+by+m=0 \\ cx+dy+n=0 \end{cases}$ 的解，a,b,m,c,d,n 作为单元格参数输入，如它们的值依次为1,1,-9,2,-2,-10 。

解题思路

二元一次方程组的解为$x=\frac{bn-dm}{ad-bc}$，为$y=\frac{cm-an}{ad-bc}$，计算后将结果记录输出即可。

注意

表达式中可以用"..."表示字符串...

	A	B	C	D
1	1	1	-9	/输入方程1的参数a,b,m
2	2	-2	-10	/输入方程2的参数c,d,n
3	=(B1*C2-B2*C1)/(A1*B2-B1*A2)			/计算x
4	=(A2*C1-A1*C2)/(A1*B2-B1*A2)			/计算y
5	>output("x="/A3/", y="/A4)			/输出结果

x=7.0, y=2.0

在SPL中输入和输出数据

- 最常用的输入数据方法是直接在网格中输入数据，利用单元格常数和单元格变量执行计算。
- SPL中的运算结果可以用单元格值直接查看，不一定需要输出。
- 在调试程序时，可以使用**output()**函数，将字串和其它类型数据输出到控制台。
- 除了直接使用单元格常量，也可以通过定义网格程序参数的方式输入数据。有关网格参数的使用，将在第6章中介绍。

3.6 习题

3.6 习题

【习题3.1】如果x的初始值为5，执行 x+=x-3 后，x的值是多少?

【习题3.2】输入3个数a,b,c，计算a+b/c的结果，结果保留3位小数。（保留3位小数用函数round(x,3))

【习题3.3】输入一个四位数，输出它的每位数字，中间用逗号分隔。

【习题3.4】根据输入的华氏温度，输出对应的摄氏温度。华氏温度F转换为摄氏温度C的公式如下:

$$C = \frac{5}{9}(F - 32).$$

【习题3.5】输入p=4.5，q=321，用程序交换它们的值。

【习题3.6】读下面的程序，请问程序的功能是什么？执行后输出的结果是什么?

	A	B	C
1	123	23	91
2	=(A1+B1+C1)/3		
3	>output("a="/A1/",b="/B1/",c="/C1/",Avg="/A2)		

3.6 习题

【习题3.1】如果x的初始值为5，执行 x+=x-3 后，x的值是多少?

	A	B
1	5	/输入x
2	>A1+=A1-3	/执行计算

或者:

	A	B
1	>x=5	/输入x
2	>x+=x-3	/执行计算

Value
7

3.6 习题

【习题3.2】输入3个数a,b,c，计算a+b/c的结果，结果保留3位小数。（保留3位小数用函数round(x,3)）

	A	B	C	D
1	5	2	3	/输入a,b,c
2	=round(A1+B1/C1, 3)			/执行计算

Value
5.667

3.6 习题

【习题3.3】输入一个四位数，输出它的每位数字，中间用逗号分隔。

	A	B	C	D	E
1	1357				/输入四位数
2	=A1\1000	=(A1%1000)\100	=(A1%100)\10	=A1%10	/执行计算
3	=A2/","/B2/","/C2/","/D2				/获得结果

Value
1,3,5,7

3.6 习题

【习题3.4】根据输入的华氏温度，输出对应的摄氏温度。华氏温度F转换为摄氏温度C的公式如下：

$$C = \frac{5}{9}(F - 32).$$

	A	B
1	95	/输入华氏温度
2	=round((A1-32)*5/9,1)	/转换摄氏温度
3	>output(A1/" C = "/A2/" F")	/输出结果

```
95 C = 35.0 F
```

3.6 习题

【习题3.5】输入p=4.5，q=321，用程序交换它们的值。

SPL如下：

	A	B
1	>p=4.5,q=321	/输入p,q
2	=p	/A2←p
3	>p=q	/p←q
4	>q=A2	/q←A2

No.	Name	Value
1	p	321
2	q	4.5

3.6 习题

【习题3.6】读下面的程序，请问程序的功能是什么？执行后输出的结果是什么？

	A	B	C
1	123	23	91
2	=(A1+B1+C1)/3		
3	>output("a="/A1/",b="/B1/",c="/C1/",Avg="/A2)		

程序用来求输入3个数的平均值，结果如下：

```
a=123,b=23,c=91,Avg=79.0
```

第4章 选择结构程序设计

4.1 选择结构和条件判断

4.2 用 if 语句实现选择结构

- 选择结构举例
- If 语句的一般形式
- If 语句的其它形式

4.3 关系运算符和关系表达式

- 关系运算符及优先次序
- 关系表达式

4.4 逻辑运算符和逻辑表达式

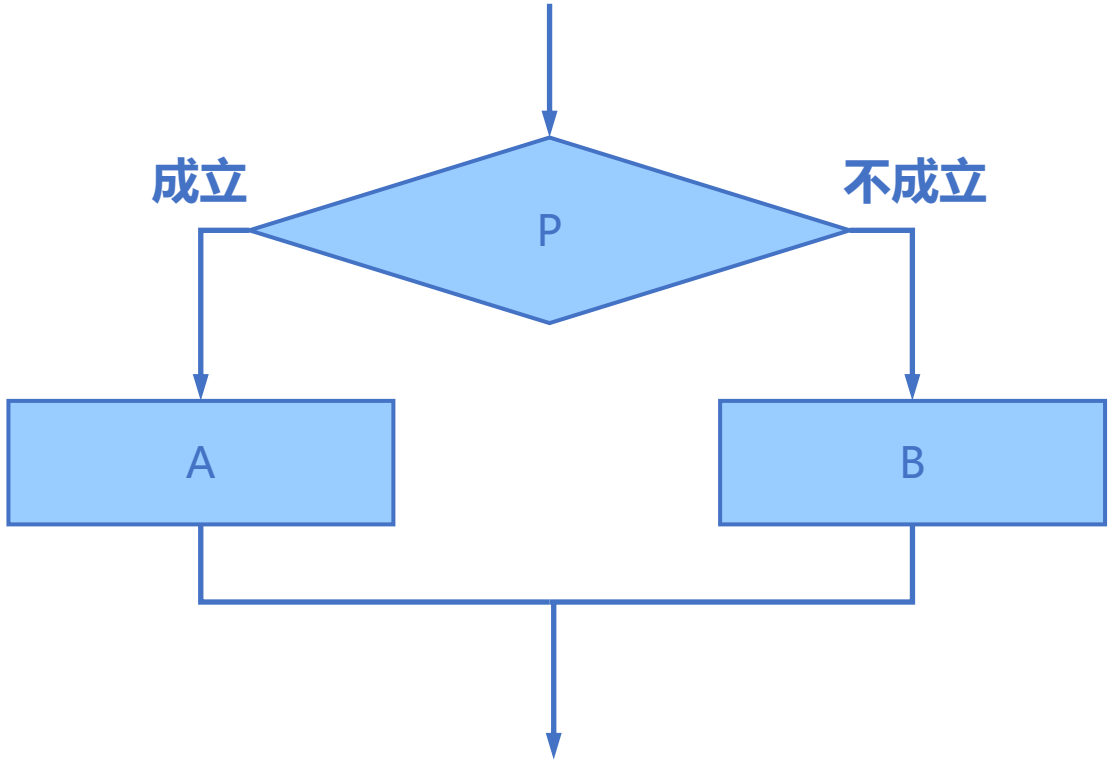
- 逻辑运算符及优先次序
- 逻辑表达式

4.5 if 函数

4.6 选择结构的嵌套

4.7 习题

4.1 选择结构和条件判断


成立
P
不成立
A
B

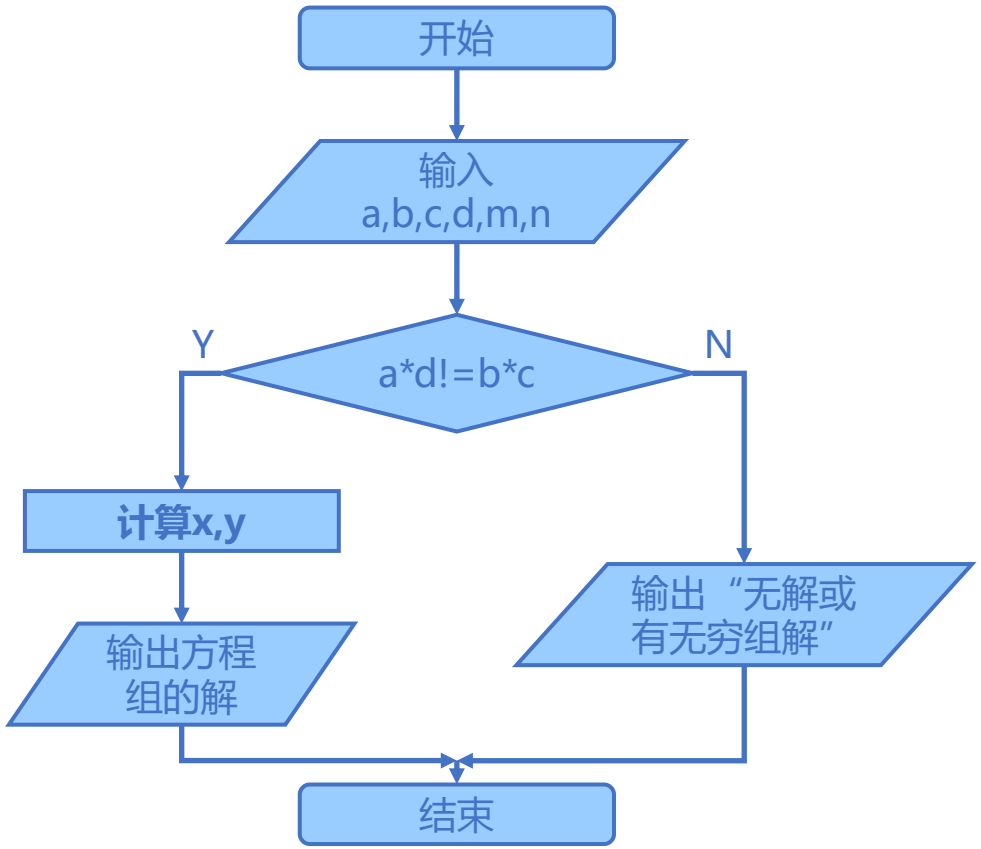
4.2 用 if 语句实现选择结构

【例4.1】例3.4中，我们研究过求二元一次方程组 $\begin{cases} ax+by+m=0 \\ cx+dy+n=0 \end{cases}$ 解的问题，实际上，在处理问题时， 推算结果$x=\frac{bn-dm}{ad-bc}$，$y=\frac{cm-an}{ad-bc}$时，需要考虑$ad\text{-}bc$为0的情况，此时有可能无解或有无数组解。

4.2 用 if 语句实现选择结构

选择结构举例

【例4.1】例3.4中，我们研究过求二元一次方程组 $\begin{cases} ax+by+m=0 \\ cx+dy+n=0 \end{cases}$ 解的问题，实际上，在处理问题时， 推算结果$x=\frac{bn-dm}{ad-bc}$， $y=\frac{cm-an}{ad-bc}$时，需要考虑$ad\text{-}bc$ 为0的情况，此时有可能无解或有无数组解。



	A	B	C	D
1	1	1	-9	/输入参数a,b,m
2	2	2	-10	/输入参数c,d,n
3	if A1*B2!=B1*A2	=(B1*C2-B2*C1)/(A1*B2-B1*A2)	=(A2*C1-A1*C2)/(A1*B2-B1*A2)	/计算x，y
4		>output("x="/B3/", y="/C3)		/输出结果
5	else	>output("No solution or infinite solutions.")		/无解或无穷解

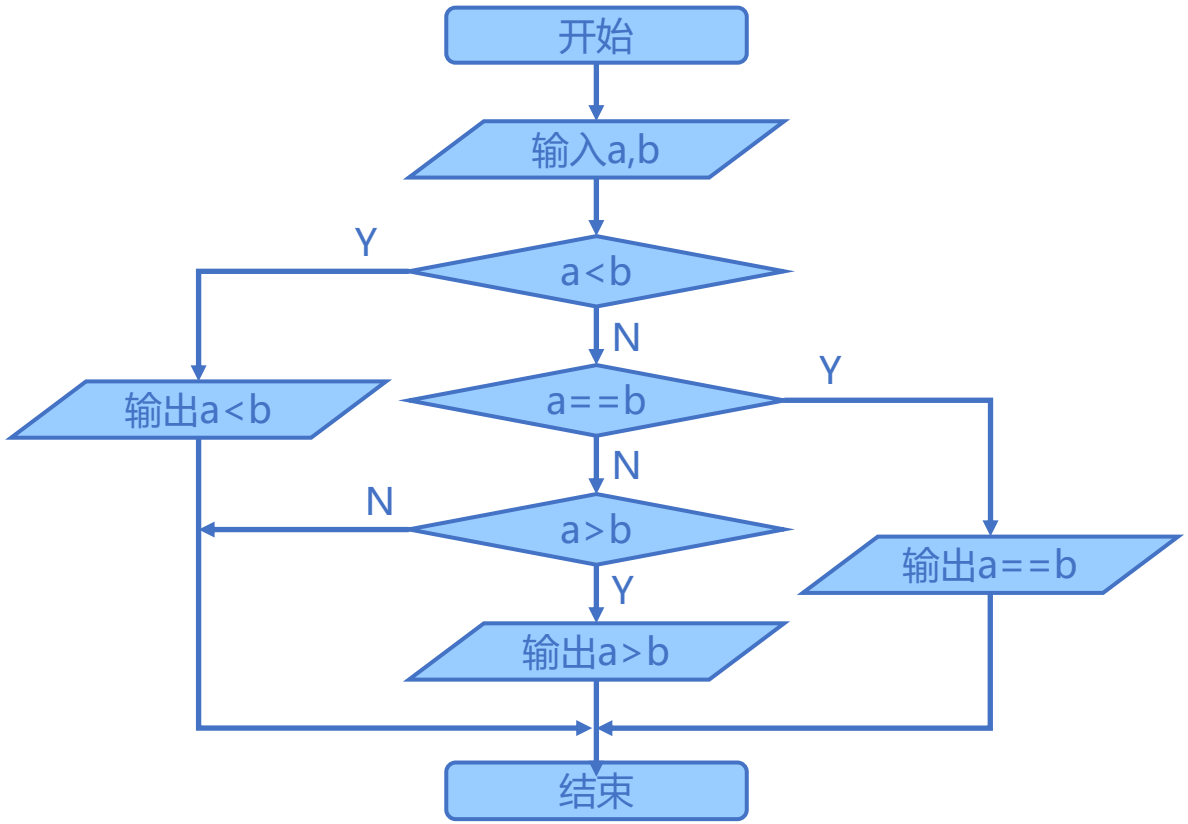
No solution or infinite solutions.

【例4.2】输入两个实数，比较它们之间的大小关系并输出结果。

【例4.2】输入两个实数，比较它们之间的大小关系并输出结果。

解题思路

判断它们之间大小关系的不同情况，在两个实数之间添加不同的符号。

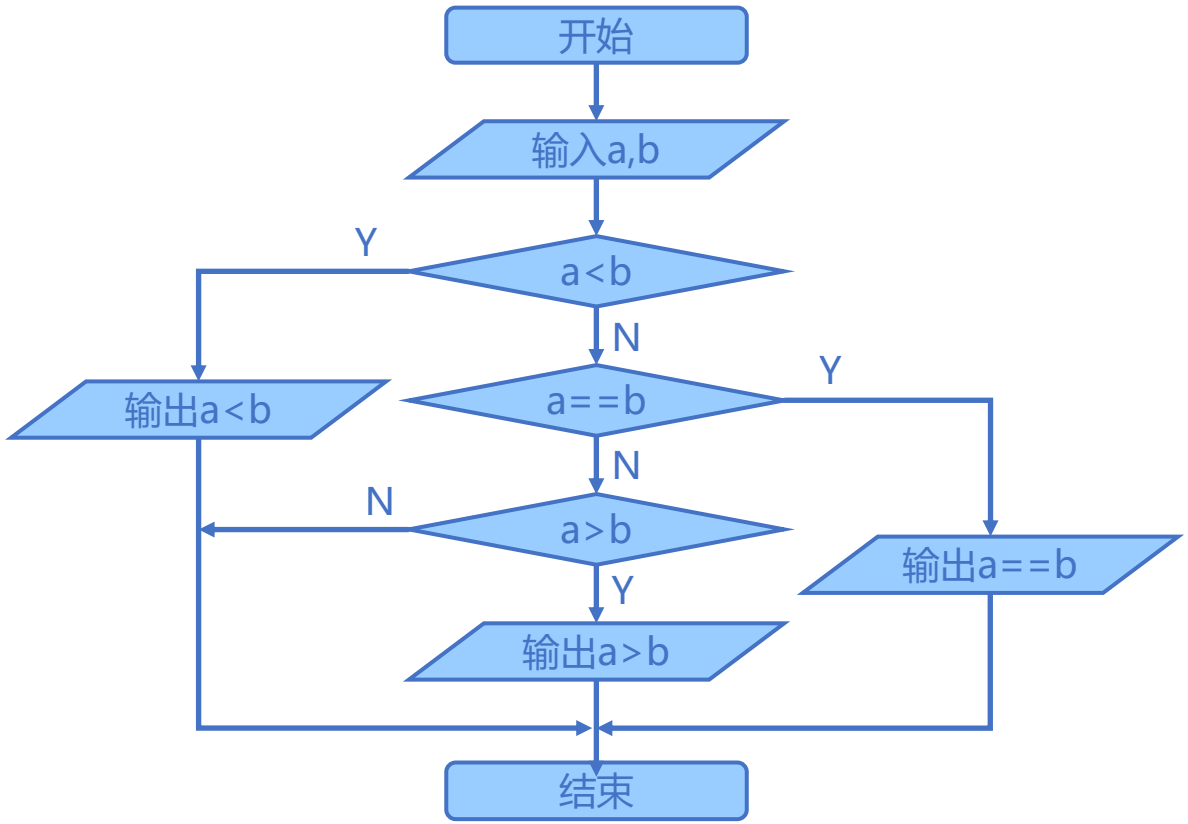


4.2 用 if 语句实现选择结构

【例4.2】输入两个实数，比较它们之间的大小关系并输出结果。

解题思路

判断它们之间大小关系的不同情况，在两个实数之间添加不同的符号。



	A	B	C
1	3.1416	3	/输入两个实数
2	if A1<B1	>output(A1/"<"/B1)	/A1是否小于B1
3	if A1==B1	>output(A1/"=="/B1)	/A1是否等于B1
4	if A1>B1	>output(A1/">"/B1)	/A1是否大于B1

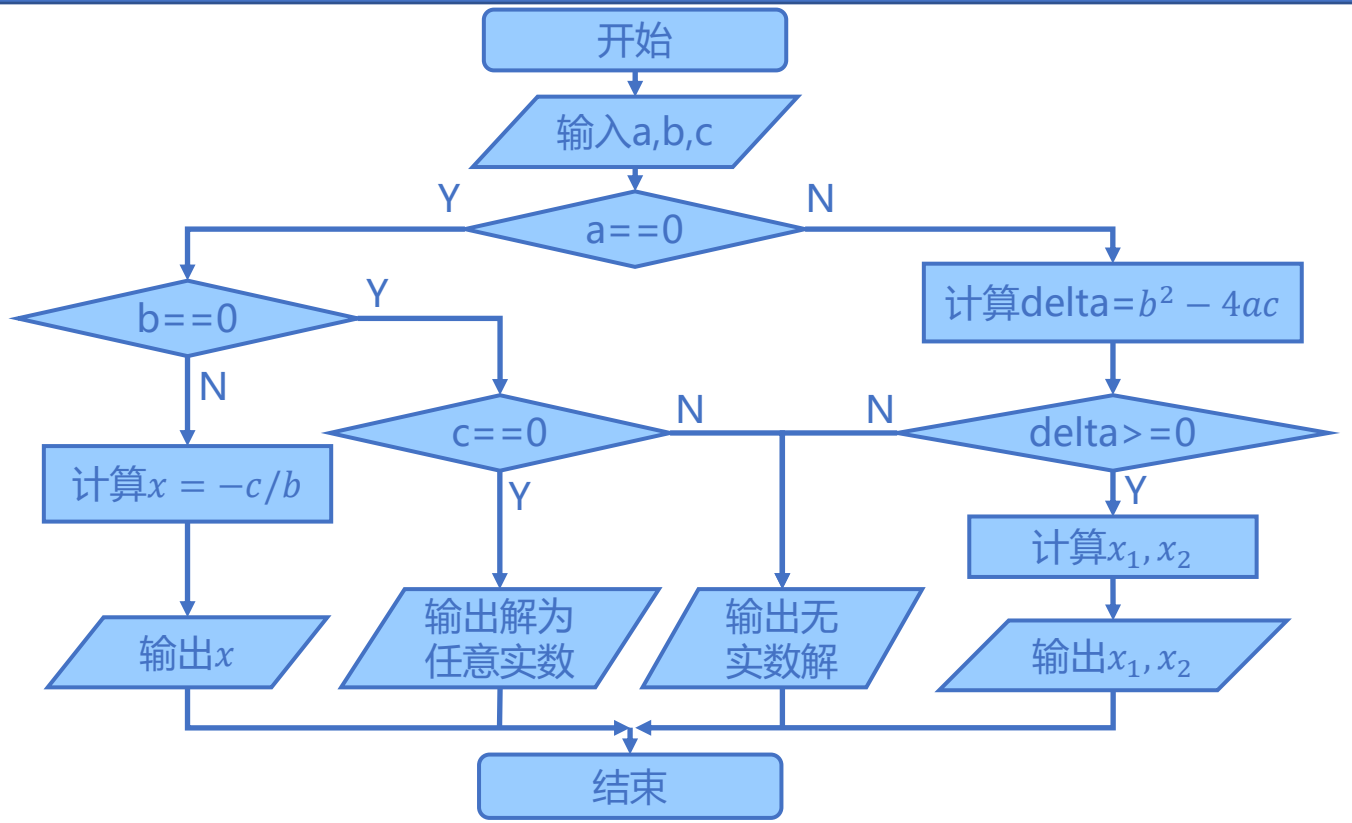
```
3.1416>3
```

【例4.3】输入参数 a, b, c，求方程 $ax^2 + bx + c = 0$ 的实数解。

【例4.3】输入参数a,b,c，求方程 $ax^2+bx+c=0$ 的实数解。

解题思路

判断a是否为0，若a为0，则继续判断b是否为0，若b为0，则继续判断c是否为0，若c为0，则得到任意实数解，若c不为0，则无实数解；当a为0，b不为0，则直接计算一元一次方程bx+c=0的实数解；当a不为0，则计算一元二次方程的delta判别式，若delta大于等于0，则计算一元二次方程的两个实数解，若delta小于0，则一元二次方程无实数解。



【例4.3】输入参数a,b,c，求方程 $ax^2+bx+c=0$ 的实数解。

	A	B	C	D	E
1	2	3	-4		/输入a,b,c
2	if A1==0	if B1==0	if C1==0	>output("This equation is solved as any real number.")	/abc均为0
3			else	>output("This equation has no real number solution.")	/无实数解
4		else	=round(-C1/B1,3)	>output("This solution of the equation is "/C4/".")	/一次方程的解
5	else	=B1*B1-4*A1*C1			/计算判别式
6		if B5>=0	=-B1/(2*A1)	=sqrt(B5)/(2*A1)	/一元二次方程
7			=round(C6+D6,3)	=round(C6-D6,3)	/计算x_1,x_2
8				>output("The roots of this equation are "/C7/" and "/D7/".")	/输出一元二次方程的解
9		else		>output("This equation has no real number solution.")	/无实根

运行结果：

```
The roots of this equation are 0.851 and -2.351.
```

A1，B1，C1中输入的方程参数可以根据需要调整。

if 语句的一般形式

- if 表达式　　代码块1
  [else　　代码块2]
- if表达式通常为关系表达式或者逻辑表达式，也可以是其它表达式。执行后，表达式的计算结果将会作为单元格的格值返回。
- 当表达式的结果为空值或false时，将执行代码块2；否则认为if条件成立，执行代码块1。选择结构执行后将继续执行if语句后面的代码。
- else子句部分是可以没有的，此时只在条件成立时执行代码块1。
- 代码块1和代码块2可以是单个SPL语句，也可以是多个SPL语句所构成的一系列代码，代码块中仍然可以继续使用if语句。

代码块

- SPL中用**代码块**来决定语句的作用范围
- if语句所在的格，是它的代码块的**起始格**，也称为**代码块的主格**。
- 如果主格下面的连续若干行，主格所在的列及其左侧均为空格，则这些行和主格所在的行一起构成主格的代码块。

	A	B	C	D
1				
2		>Code begin		
3				
4				
5				
6	Not null	Not null		

- 代码块的范围直到某一行主格所在列或其左边的任何一个单元格非空为止。

单行的if...else...语句

- if 表达式　　代码块1　　else　　代码块2

	A	B	C	D
1	-14			
2	if A1>=0	>B1=A1	else	>B1=-A1

- 单行的if...else...语句，它们各自的代码块可以由多个SPL语句构成，但应该全部写在一行中。

else if 子句

- if 表达式1　　代码块1
  else if 表达式2　　代码块2
  else if 表达式3　　代码块3
  [else　　代码块0]
- 多条else if子句将逐条判断，直到某个表达式n的条件成立，执行代码块n；或所有条件均不成立，至else执行代码块0。多重选择结构执行之后，也会继续执行if语句后面的代码。
- else子句部分是可以没有的，此时若所有条件均不成立则继续执行if语句外的代码。

4.3 关系运算符和关系表达式

关系运算符

运算符	含义	举例	结果
>	大于	*a*>*b*	如果 *a*大于*b* 为true，否则为false
>=	大于或等于(不小于)	*a*>=*b*	如果 *a*大于*b* 或 *a* 等于*b* 为true，否则为false
<	小于	*a*<*b*	如果 *a*小于*b* 为true，否则为false
<=	小于或等于(不大于)	*a*<=*b*	如果 *a*小于*b* 或 *a* 等于*b* 为true，否则为false
==	等于	*a*==*b*	如果 *a*与*b* 相等为true，否则为false
!=	不等于	*a*!=*b*	如果 *a*与*b* 不等为true，否则为false

关系运算符的优先次序

- 在关系运算符中，运算时的优先级分为两个级别：大于、大于等于、小于、小于等于这4个运算符优先级较高，会先行计算；等于、不等于这2个运算符优先级较低。
- 在计算优先级相同的关系运算符时，将按照从左向右的顺序。
- 关系运算符的优先级低于算术运算符。
- 关系运算符的优先级低于非运算符(!)，但高于其它逻辑运算符。

关系表达式

- 用关系运算符将两个数值或者数值表达式连接起来的式子，称为关系表达式。
- 关系表达式的结果是布尔值，根据关系表达式是否成立得到true或false。
- SPL的关系表达式中，不同类型的数据不能比较大小，但可以比较是否相等。

4.4 逻辑运算符和逻辑表达式

逻辑运算符

运算符	含义	举例	结果
!	逻辑非运算符	!a	如果a为true，结果为false；如果a为false，结果为true
&&	逻辑与运算符	a&&b	如果a与b全部为true，结果为true；否则结果为false
||	逻辑或运算符	a||b	如果a与b中至少有1个为true，结果为true；否则结果为false

a	*b*	*!a*	*!b*	*a&&b*	*a||b*
true	true	false	false	true	true
true	false	false	true	false	true
false	true	true	false	false	true
false	false	true	true	false	false

逻辑运算符的优先次序

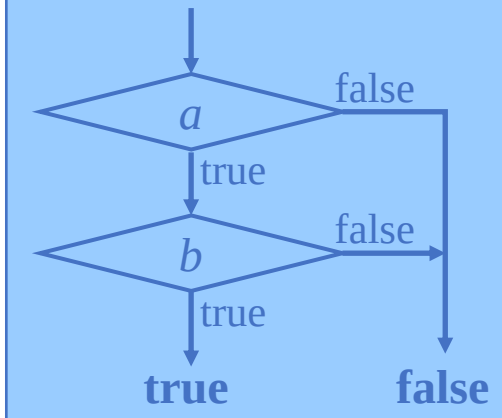
- 逻辑非运算符是单目运算，优先级比所有双目运算符都要高。逻辑非运算符的优先级低于正号和负号运算符，高于其它算术运算符和关系运算符。
- 逻辑与和逻辑或运算符的优先级较低，低于算术运算符和关系运算符。
- 逻辑与的优先级高于逻辑或，两者同时存在时先计算与再计算或。

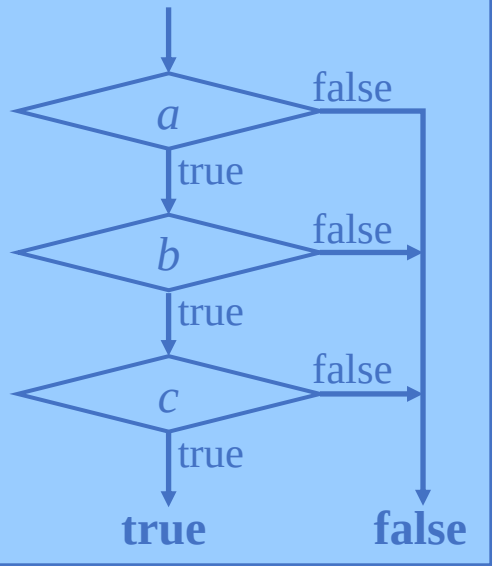
逻辑表达式

- 用逻辑运算符将关系表达式或者其他布尔值操作数连接起来的式子就是逻辑表达式。
- 逻辑表达式的结果也是布尔值true或false。
- SPL的逻辑表达式中，逻辑运算符使用的操作数都应该是布尔型的true或者false，其它类型的任何数值都会被作为true参与计算，空值null会被作为false参与计算。

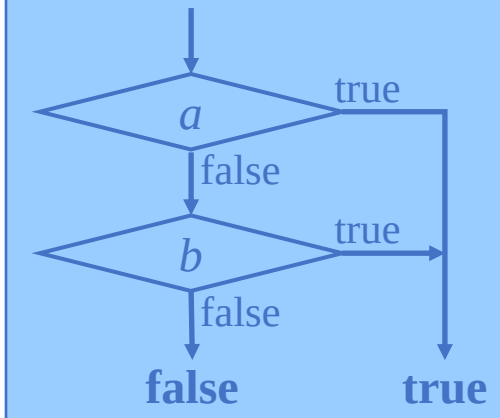
在逻辑表达式中，可以使用多个结果为布尔值的关系表达式或逻辑表达式。
在逻辑表达式计算时，这些表达式并不一定都要计算出结果。

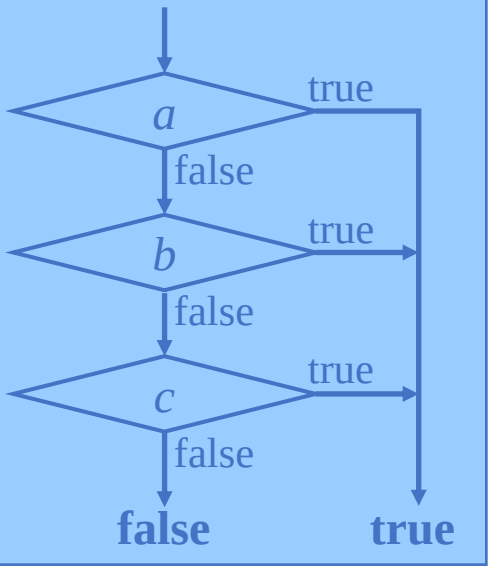
- 计算 $a\&\&b$ 时，若 a 为true时，才需要继续判断 b 的值。在计算 $a\&\&b\&\&c$ 时，若 a 为true时，才需要继续判断 b 的值；若 a 和 b 均为true时，才需要继续判断 c 的值。



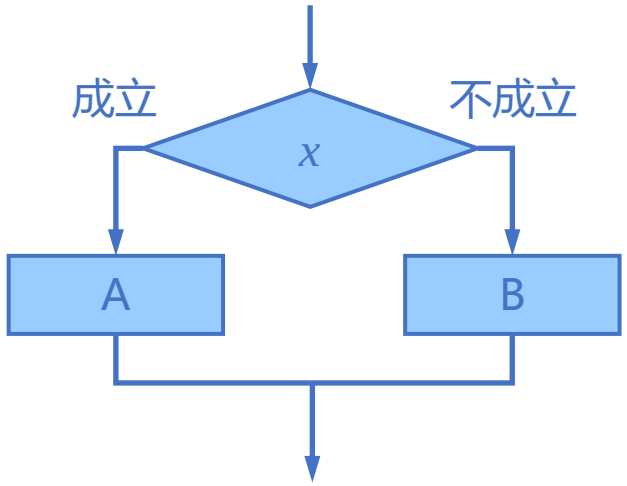


- 计算 $a \| b$ 时，若 a 为false时，才需要继续判断 b 的值。在计算 $a \| b \| c$ 时，若 a 为false时，才需要继续判断 b 的值；若 a 和 b 均为false时，才需要继续判断 c 的值。





逻辑表达式和关系表达式返回的结果都是布尔值true或false，因此它们都可以用在if语句的条件中，当结果为真true时，条件成立；结果为false时，条件不成立。



- 当*x*为布尔值时，当*x*为true时条件成立，执行A；　*x*为false时条件不成立，执行B。
- 当*x*为关系表达式时，当结果为true时条件成立，执行A；结果为false时条件不成立，执行B。
- 当*x*为逻辑表达式时，当结果为true时条件成立，执行A；结果为false时条件不成立，执行B。
- 当*x*的计算结果为非布尔类型的数值时，如任何整数、字符等，将认为条件成立，执行A；如果结果为空值时，认为条件不成立，执行B。

4.5 If 函数

4.5 if 函数

	A	B	C
1	3.1416	3	/输入两个实数
2	if A1<B1	>max=B1	/如A1小于B1，max←B1
3	else	>max=A1	/否则max←A1

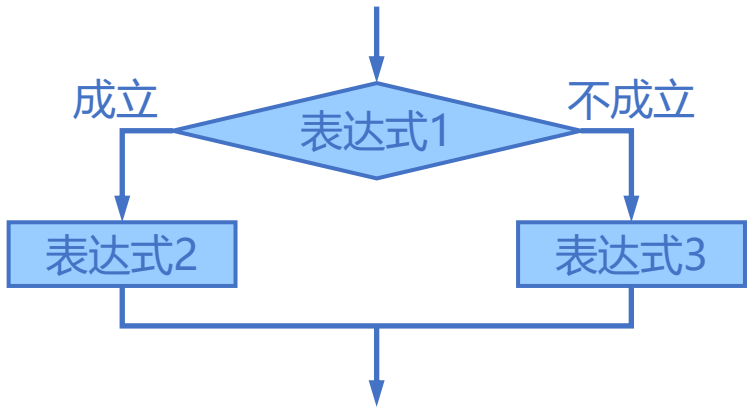
>max=if(A1<B1,B1,A1)

或>if(A1<B1, max=B1, max=A1)

No.	Name	Value
1	max	3.1416

if函数

- if(表达式1, 表达式2, 表达式3)
- 一般格式的if函数有3个参数，常用表达式来作为函数的参数。计算时先求解表达式1，若为false则执行表达式3并返回其结果；否则执行表达式2并返回结果。
- 当表达式1成立时，只会执行表达式2，表达式3并不会被执行；当表达式1不成立时，也同样只会执行表达式3，表达式2不会被执行。



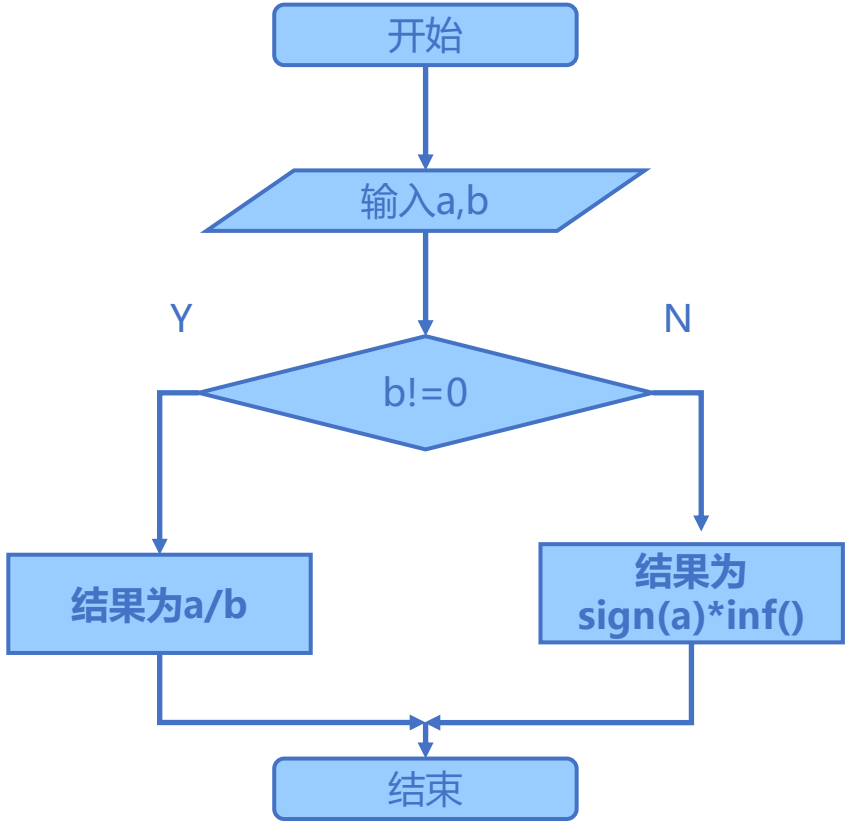
【例4.4】输入a, b，计算 a 除以 b 的值，结果保留3位小数。

4.5 if 函数

【例4.4】输入a,b，计算 a 除以 b 的值，结果保留3位小数。

解题思路

先需要判断 b 是否为0，如果为0时，用sign()函数获得 a 的正负和inf()表示的无穷大来计算结果。计算时可以直接用if函数来处理。

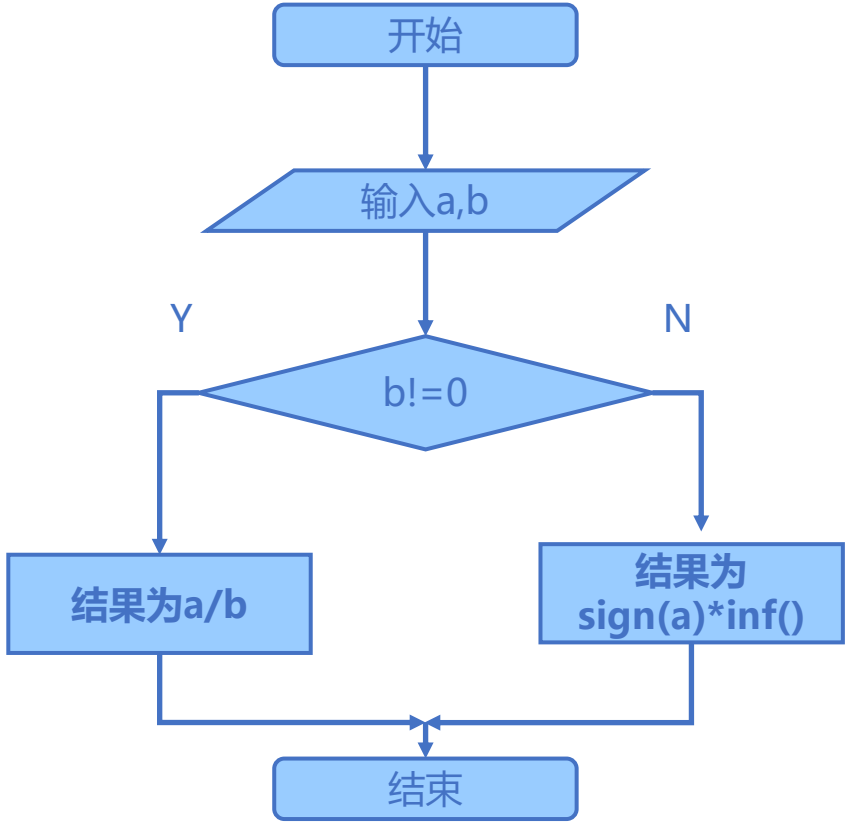


4.5 if 函数

【例4.4】输入a,b，计算 a 除以 b 的值，结果保留3位小数。

解题思路

先需要判断 b 是否为0，如果为0时，用sign()函数获得 a 的正负和inf()表示的无穷大来计算结果。计算时可以直接用if函数来处理。



	A	B	C
1	-4	0	/输入参数a,b
2	=if(B1!=0, round(A1/B1,3), sign(A1)*inf())		/计算a/b, 考虑b是否为0

Value
-Infinity

虽然A1/B1无法正常计算，但由于在if表达式中只会根据表达式1的结果来求解对应的表达式，因此上面的计算并不会报错。

4.5 if 函数

与 if 语句的形式类似，if 函数也有多种格式：

- if(表达式1, 表达式2)
  当表达式1的结果为true时，会计算表达式2；否则返回空值。

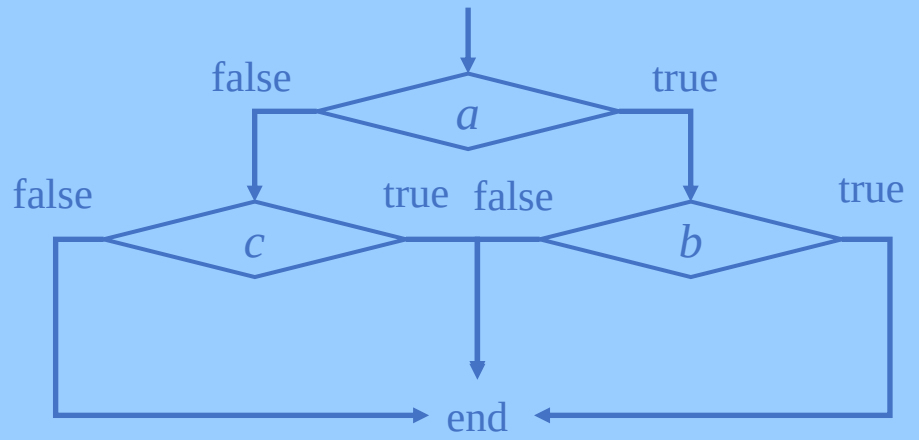
- if(表达式1a, 表达式1b, 表达式2a, 表达式2b, ..., 表达式na, 表达式nb, 表达式0)
  当表达式1a的结果为true时，计算表达式1b并返回结果；否则继续计算表达式2a，若其结果为true则计算表达式2b并返回结果；否则继续计算表达式3a,...如果直到表达式na都无法满足，则计算表达式0并返回结果。这种格式的if函数类似多个 else if 语句的情况。

4.6 选择结构的嵌套

4.6 选择结构的嵌套

	A	B	C	D
1	if a			/外层if判断
2		if b		/嵌套if判断
3		else		/嵌套else语句
4	else			/外层else语句
5		if c		/嵌套if判断
6		else		/嵌套else语句

- 计算外层if a后，若a为true，则执行外层if后的代码，再执行if b，若b为true，则执行嵌套if b后的代码，若b为false，则执行与if b对应的嵌套else语句；若a为false，则跳过外层if a代码块的内容，直接执行外层else语句，再判断if c...



4.6 选择结构的嵌套

	A	B	C	D
1	if a			/外层if判断
2		if b		/嵌套if判断
3		else		/嵌套else语句
4	else			/外层else语句
5		if c		/嵌套if判断
6		else		/嵌套else语句

- 配对的if与else语句需要对齐。
- if和else可以有各自的代码块，当条件成立时执行if后的代码块，否则执行else后的代码块。

4.7 习题

4.7 习题

【习题4.1】输入两个数，将它们按照从大到小的顺序输出。

【习题4.2】输入一个字符，将其转换后输出，转换时，小写字母需转换为大写，其它不变。

【习题4.3】在代码中直接写入某人的体重(单位kg)和身高(单位m)，并计算出他的体重指数bmi=体重/(身高^2)。简单判断他的体型标准，当bmi小于18.5时为“偏瘦”，在18.5到小于25之间为“标准”，在25和27之间时显示“超重”，大于或等于27时显示“肥胖”。

【习题4.4】根据父母的遗传因素来预测孩子成年后的身高，可以使用下面的估算公式：设fH为父亲身高，mH为母亲身高，则男孩成年后身高=(fH+mH)*0.54，女孩成年后身高=(fH*0.97+mH)/2. 请根据输入的父母身高和孩子性别Gender，返回其预测身高，单位为cm，保留一位小数。

【习题4.5】用if函数解决习题4.1和4.2。

【习题4.6】读下面的程序，请问执行后输出的结果是什么？为什么C1与C2输出的结果不同？

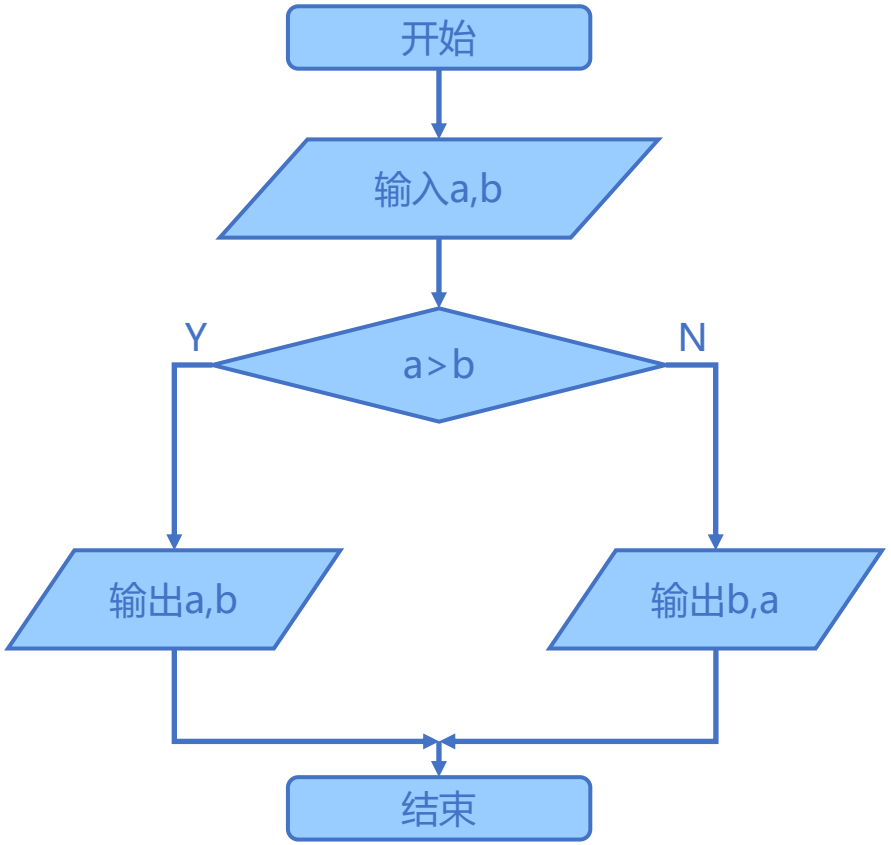
	A	B	C
1	>(a=1,b=1,c=1)	=(a=a+1)&&(b=b+1)||(c=c+1)	>output("a="/a/",b="/b/",c="/c)
2	>(a=1,b=1,c=1)	=(a=a+1)||(b=b+1)&&(c=c+1)	>output("a="/a/",b="/b/",c="/c)

【习题4.7】输入一个八位数字birthday，如1980年1月20日输入19800120，要求出生年份大于1899且小于2020，判断日期输入是否正确（true/false）。可以使用mid()和days()函数。如获取字串中的月份可以用int(mid(birthday, 5, 2))，判断1980年2月有多少天可以用days(date(1980,2,1))。

【习题4.1】输入两个数，将它们按照从大到小的顺序输出。

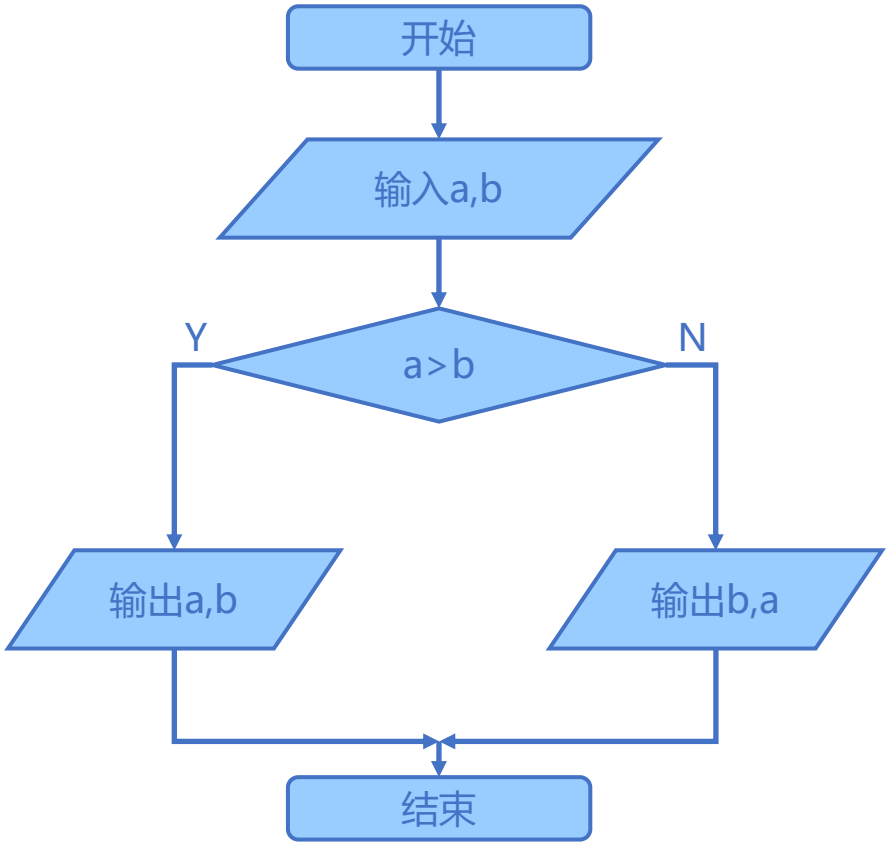
4.7 习题

【习题4.1】输入两个数，将它们按照从大到小的顺序输出。



4.7 习题

【习题4.1】输入两个数，将它们按照从大到小的顺序输出。



	A	B	C
1	10	15	/输入a, b
2	if A1>B1	>output(A1/", "/B1)	/输出a, b
3	else	>output(B1/", "/A1)	/输出b, a

结果(举例)：

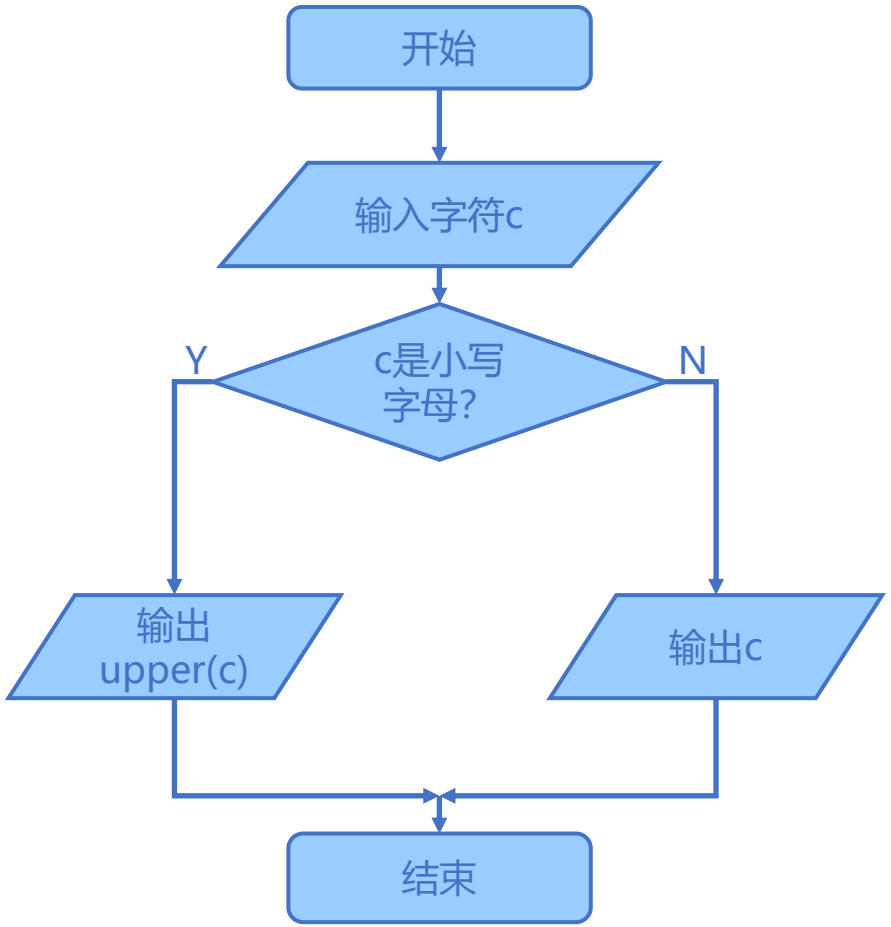
```
15, 10
```

4.7 习题

【习题4.2】输入一个字符，将其转换后输出，转换时，小写字母需转换为大写，其它不变。

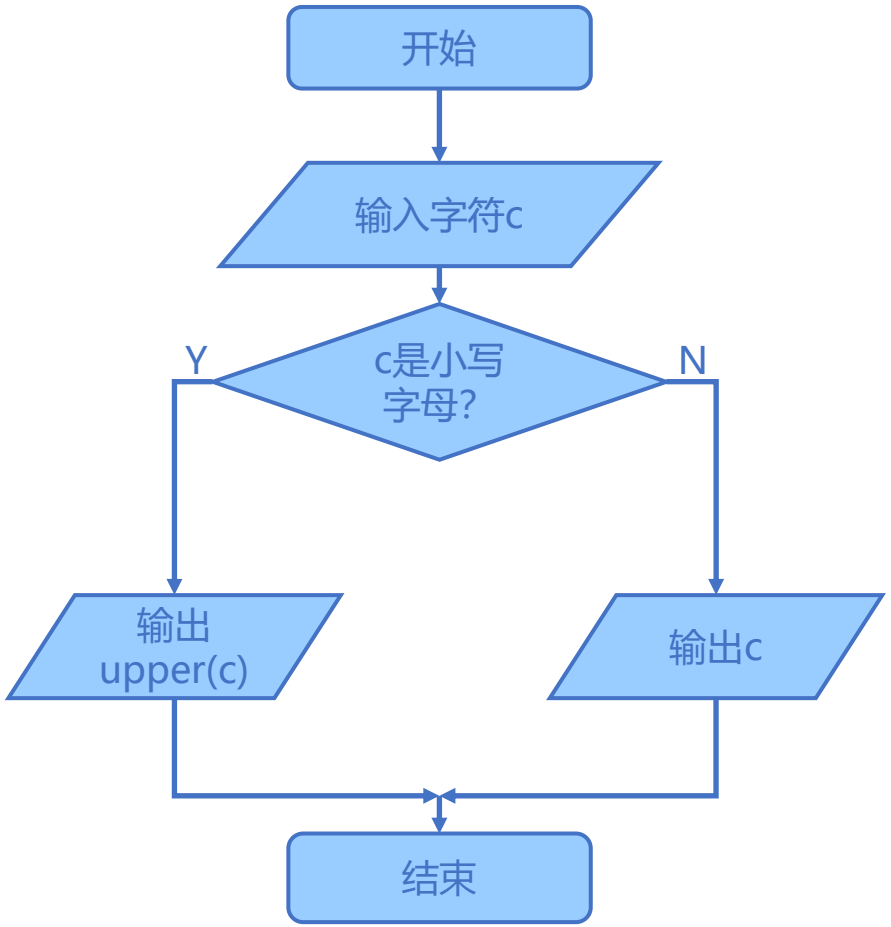
4.7 习题

【习题4.2】输入一个字符，将其转换后输出，转换时，小写字母需转换为大写，其它不变。



4.7 习题

【习题4.2】输入一个字符，将其转换后输出，转换时，小写字母需转换为大写，其它不变。



	A	B	C
1	b		/输入字符c
2	if A1>="a" && A1<="z"	>output(upper(A1))	/小写字母转为大写输出
3	else	>output(A1)	/直接输出字符

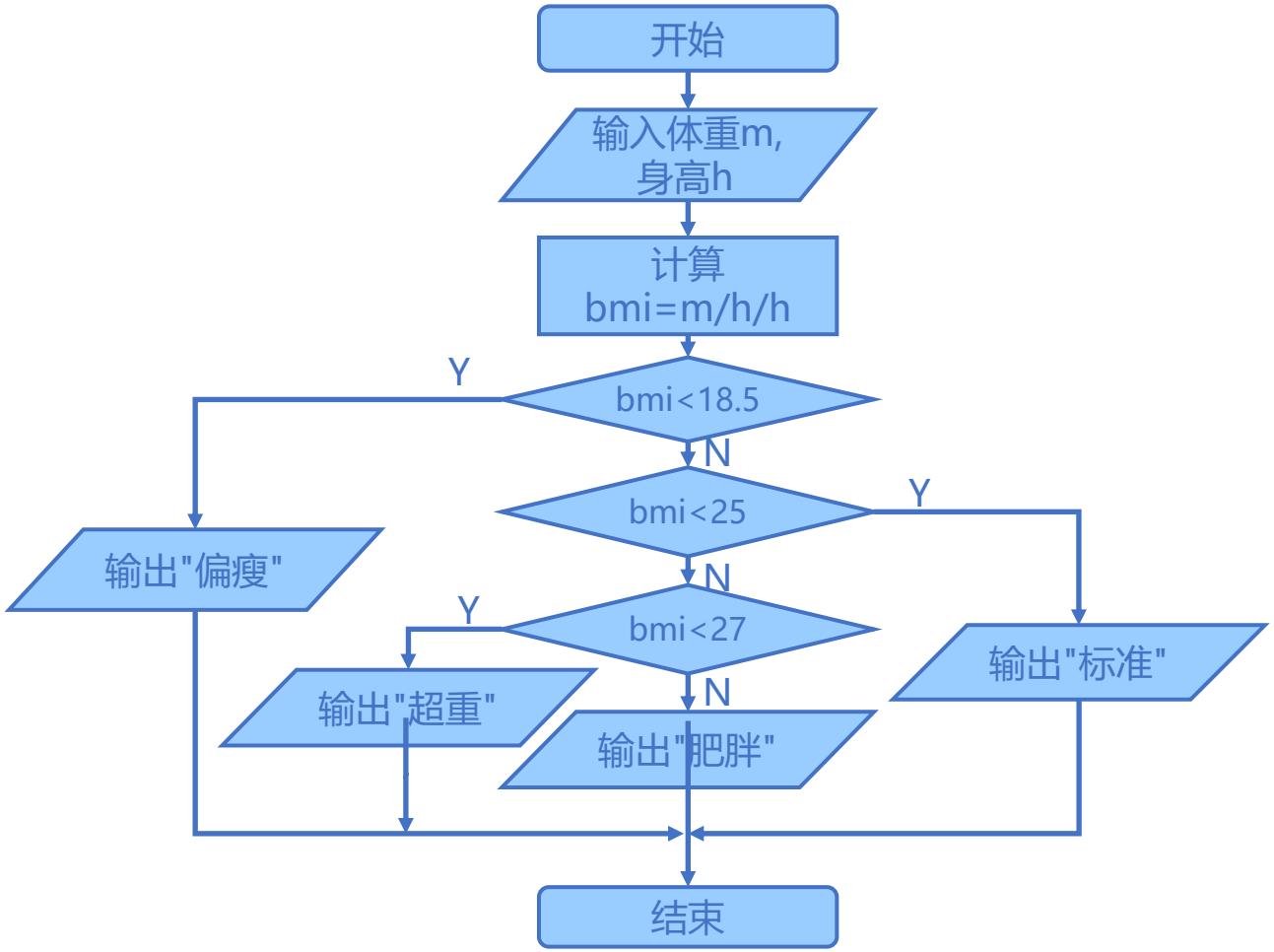
结果(举例)：

4.7 习题

【习题4.3】在代码中直接写入某人的体重(单位kg)和身高(单位m)，并计算出他的体重指数bmi=体重/(身高^2)。简单判断他的体型标准，当bmi小于18.5时为“偏瘦”，在18.5到小于25之间为“标准”，在25和27之间时显示“超重”，大于或等于27时显示“肥胖”。

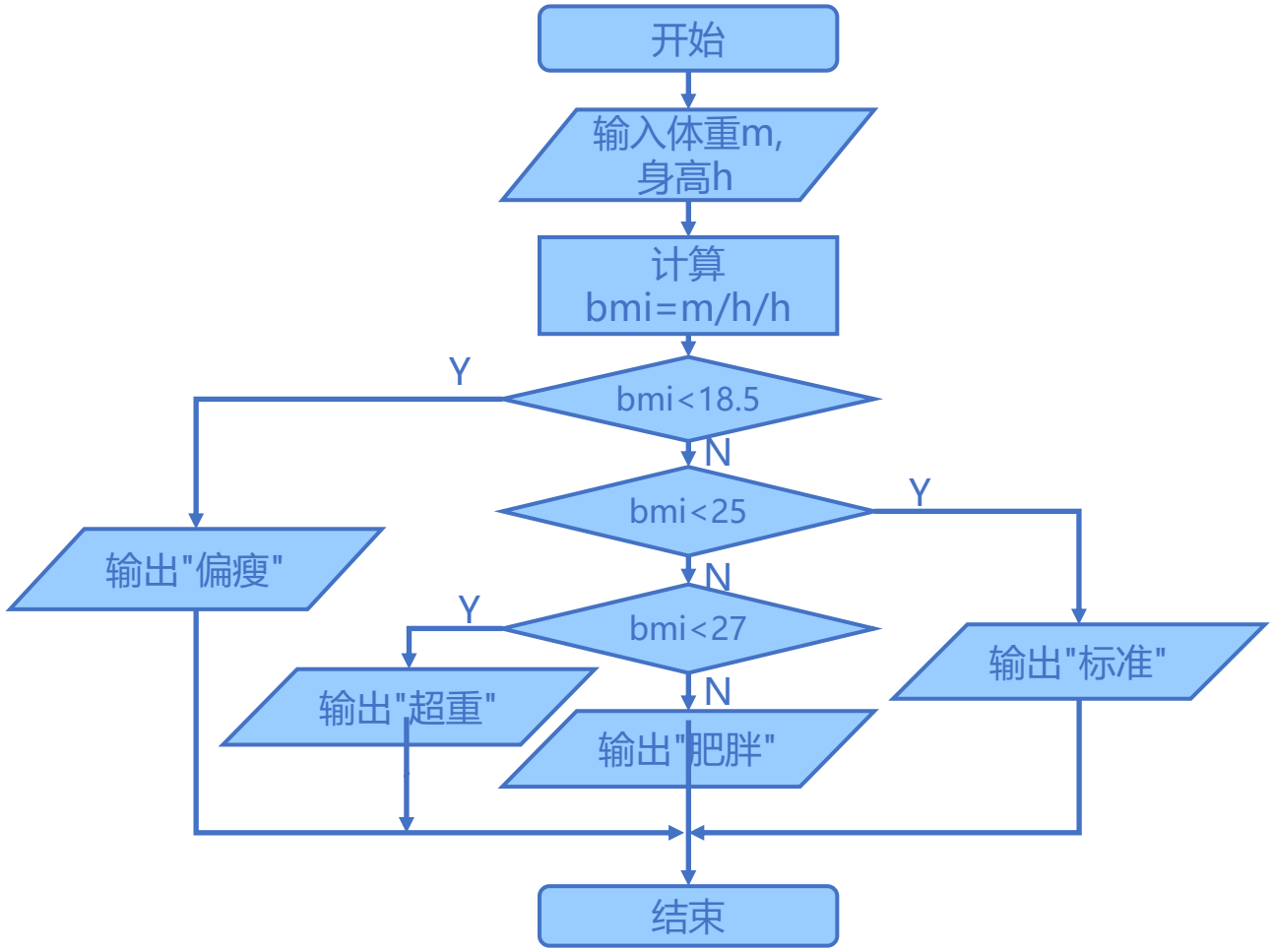
4.7 习题

【习题4.3】在代码中直接写入某人的体重(单位kg)和身高(单位m)，并计算出他的体重指数bmi=体重/(身高^2)。简单判断他的体型标准，当bmi小于18.5时为“偏瘦”，在18.5到小于25之间为“标准”，在25和27之间时显示“超重”，大于或等于27时显示“肥胖”。



4.7 习题

【习题4.3】在代码中直接写入某人的体重(单位kg)和身高(单位m)，并计算出他的体重指数bmi=体重/(身高^2)。简单判断他的体型标准，当bmi小于18.5时为“偏瘦”，在18.5到小于25之间为“标准”，在25和27之间时显示“超重”，大于或等于27时显示“肥胖”。



	A	B	C
1	80	1.75	/输入m, h
2	>bmi=A1/B1/B1		/计算bmi
3	if bmi<18.5	>output("thin")	/是否偏瘦
4	else if bmi<25	>output("fit")	/是否标准
5	else if bmi<27	>output("overweight")	/是否超重
6	else	>output("obese")	/是否肥胖

结果(举例)：

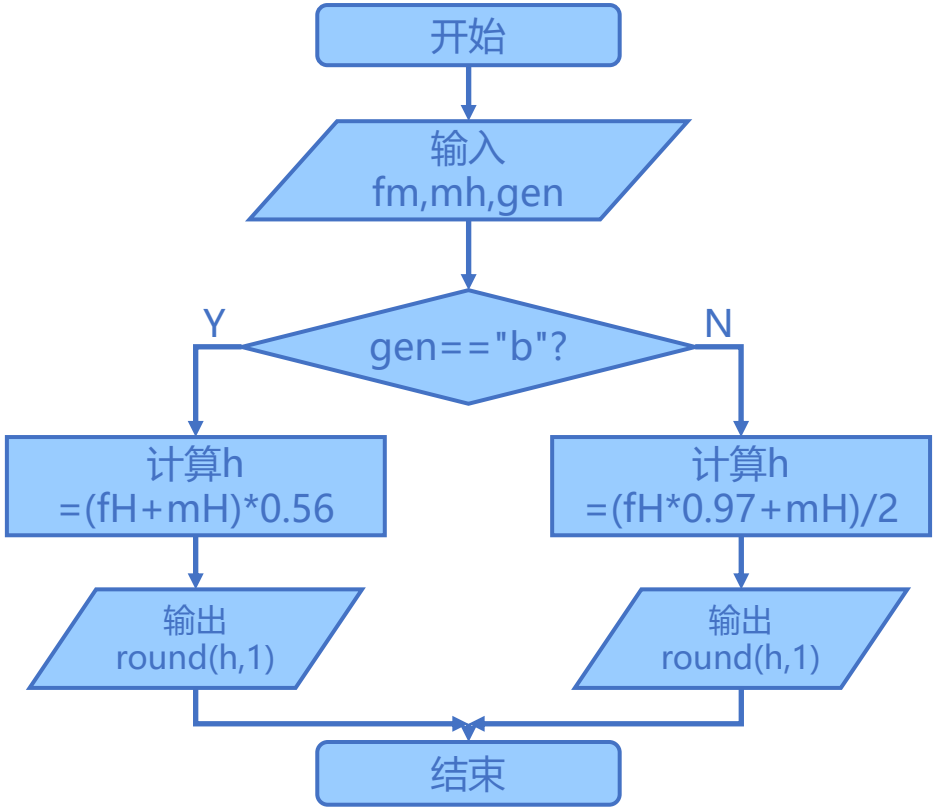
overweight

4.7 习题

【习题4.4】根据父母的遗传因素来预测孩子成年后的身高，可以使用下面的估算公式：设fH为父亲身高，mH为母亲身高，则男孩成年后身高=(fH+mH)*0.54，女孩成年后身高=(fH*0.97+mH)/2.请根据输入的父母身高和孩子性别Gender，返回其预测身高，单位为cm，保留一位小数。

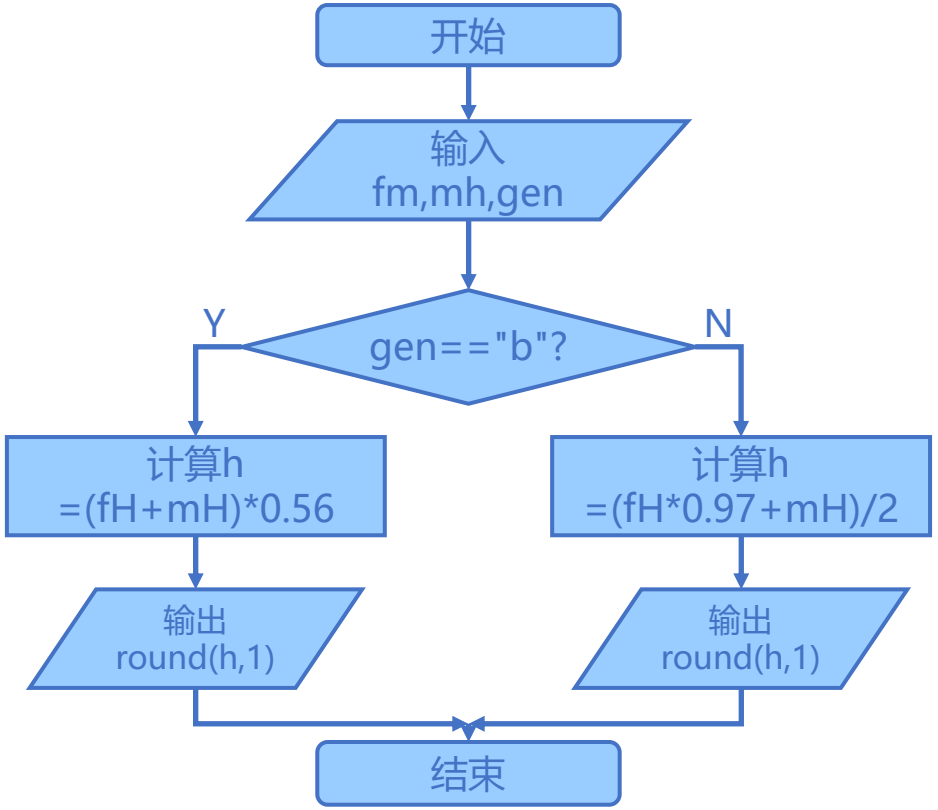
4.7 习题

【习题4.4】根据父母的遗传因素来预测孩子成年后的身高，可以使用下面的估算公式：设fH为父亲身高，mH为母亲身高，则男孩成年后身高=(fH+mH)*0.54，女孩成年后身高=(fH*0.97+mH)/2.
请根据输入的父母身高和孩子性别Gender，返回其预测身高，单位为cm，保留一位小数。



4.7 习题

【习题4.4】根据父母的遗传因素来预测孩子成年后的身高，可以使用下面的估算公式：设fH为父亲身高，mH为母亲身高，则男孩成年后身高=(fH+mH)*0.54，女孩成年后身高=(fH*0.97+mH)/2. 请根据输入的父母身高和孩子性别Gender，返回其预测身高，单位为cm，保留一位小数。



	A	B	C
1	177	157	/输入fH, mH
2	b		/输入gen
3	if A2=="b"	=(A1+B1)*0.54	/男孩
4		>output(round(B3,1))	/输出预测身高
5	else	=(A1*0.97+B1)/2	/女孩
6		>output(round(B5,1))	/输出预测身高

结果(举例)：

180.4

4.7 习题

【习题4.5】用if函数解决习题4.1和4.2。

4.1 输入两个数，将它们按照从大到小的顺序输出。

4.2 输入一个字符，将其转换后输出，转换时，小写字母需转换为大写，其它不变。

4.7 习题

【习题4.5】用if函数解决习题4.1和4.2。

4.1 输入两个数，将它们按照从大到小的顺序输出。

	A	B	C
1	10	15	/输入a, b
2	=if(A1>B1, A1/'', ''/B1, B1/'', ''/A1)		/判断顺序
3	>output(A2)		/输出结果

结果(举例)：

4.2 输入一个字符，将其转换后输出，转换时，小写字母需转换为大写，其它不变。

	A	B
1	b	/输入字符c
2	=if(A1>=''a''&&A1<=''z'', upper(A1), A1)	/转换字符
3	>output(A2)	/输出结果

结果(举例)：

4.7 习题

【习题4.6】读下面的程序，请问执行后输出的结果是什么？为什么C1与C2输出的结果不同？

	A	B	C
1	>(a=1,b=1,c=1)	=(a=a+1)&&(b=b+1)||(c=c+1)	>output("a="/a/",b="/b/",c="/c)
2	>(a=1,b=1,c=1)	=(a=a+1)||(b=b+1)&&(c=c+1)	>output("a="/a/",b="/b/",c="/c)

4.7 习题

【习题4.6】读下面的程序，请问执行后输出的结果是什么？为什么C1与C2输出的结果不同？

	A	B	C
1	>(a=1,b=1,c=1)	=(a=a+1)&&(b=b+1)||(c=c+1)	>output("a="/a/",b="/b/",c="/c)
2	>(a=1,b=1,c=1)	=(a=a+1)||(b=b+1)&&(c=c+1)	>output("a="/a/",b="/b/",c="/c)

结果：

```
a=2,b=2,c=1
a=2,b=1,c=1
```

4.7 习题

【习题4.6】读下面的程序，请问执行后输出的结果是什么？为什么C1与C2输出的结果不同？

	A	B	C
1	>(a=1,b=1,c=1)	=(a=a+1)&&(b=b+1)||(c=c+1)	>output("a="/a/",b="/b/",c="/c)
2	>(a=1,b=1,c=1)	=(a=a+1)||(b=b+1)&&(c=c+1)	>output("a="/a/",b="/b/",c="/c)

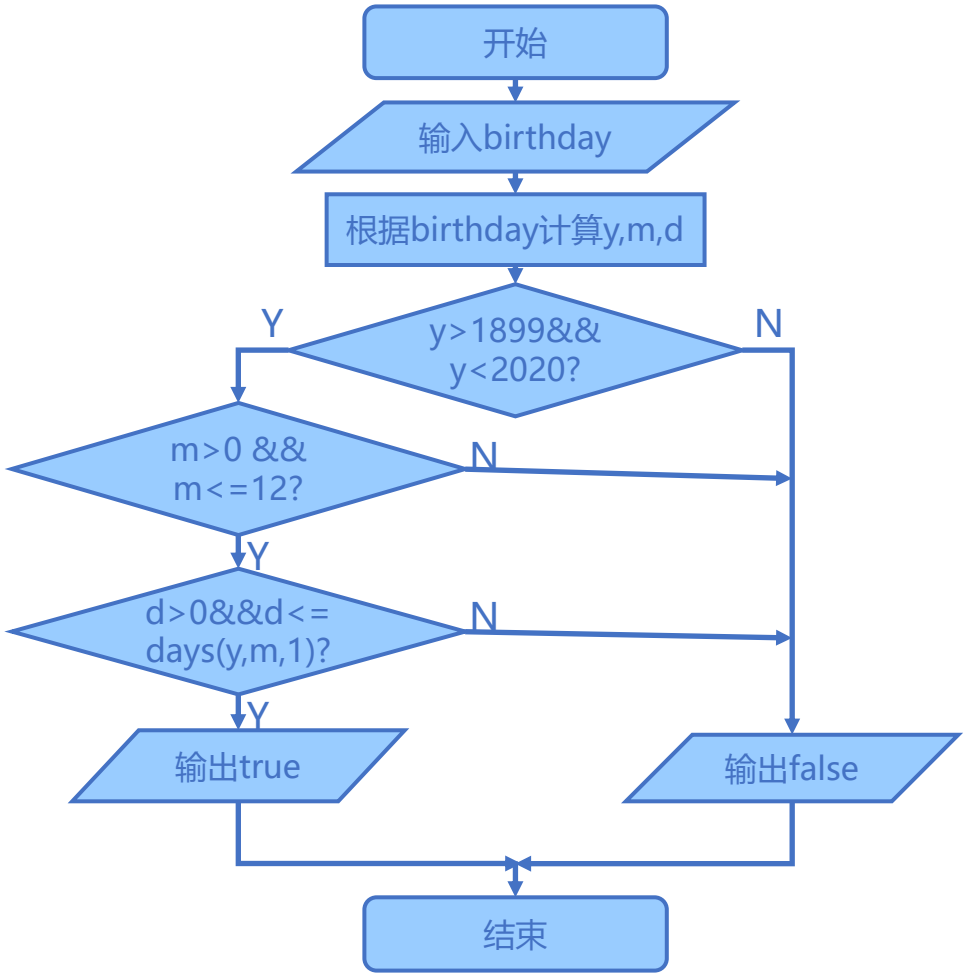
结果：

```
a=2,b=2,c=1
a=2,b=1,c=1
```

- 与(&&)的优先级高于或(||)
- &&和||的计算不需要全部执行，如a&&b，当a为false时就不会再计算b了，直接获得结果false。类似的，计算a||b时，当a为true时也不会再计算b，直接获得结果true。
- 当与或计算混合时：
  a&&b||c相当于(a&&b)||c，计算时会先计算a，如果a为false则不必再计算b，得到a&&b结果为false，只需再计算c，其结果即为最终结果；若a为true，则仍然需要计算b，b为true则结果为true，b为false则结果仍然需要计算c。例如B1中，由于a=a+1,b=b+1,c=c+1这三个表达式结果都是整数，相当于true，所以计算前两个后就可获得结果。这样，也就只有a和b被执行了加1.
  a||b&&c相当于a||(b&&c)，计算时仍然会先计算a，如果a为true，则不必再计算其它表达式，最终结果就是true；如果a为false，则再继续计算b&&c，先计算b，b为true时才会计算c。在本例中，B2中的表达式在计算a=a+1后就得到了true，不需要再计算后面两个式子了，也就只有a被加1。

4.7 习题

【习题4.7】输入一个八位数字birthday，如1980年1月20日输入19800120，要求出生年份大于1899且小于2020，判断日期输入是否正确（true/false）。可以使用mid()和days()函数。如获取字串中的月份可以用int(mid(birthday, 5, 2))，判断1980年2月有多少天可以用days(date(1980,2,1))。



	A	B	C	D
1	'19900229			/输入birthday，转为字符串
2	>y=int(mid(A1,1,4))	>m=int(mid(A1,5,2))	>d=int(mid(A1,7,2))	/计算出y,m,d
3	if y>1899 && y<2020			/年份是否满足要求
4		if m>0 && m<13	=days(date(y,m,1))	/月份是否满足要求
5			>output(d>0 && d<=C4)	/输出日期是否满足要求
6		else	>output(false)	/月份不正确，输出false
7	else	>output(false)		/年份不正确，输出false

结果(举例)：

false

第5章 循环结构程序设计

5.1 对循环结构的需求

5.2 用for x语句实现循环结构

- for x语句的一般形式
- 指定次数的循环
- break和next语句

5.3 指定循环范围的for语句

- for a, b语句
- for a, b, s语句

5.4 循环结构的嵌套

- 循环结构结构的嵌套
- 在多层循环中使用break和next

5.5 循环结构举例

5.6 习题

5.1 对循环结构的需求

5.1 对循环结构的需求

为什么需要循环结构?

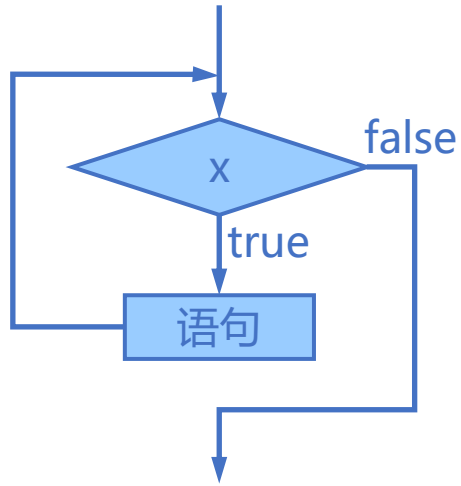
- 在解决问题时，经常需要重复执行特定的一系列计算或操作
- 在重复执行程序时，每次执行的流程相同，但是每次使用的数据或者执行的具体操作可能有区别。
- 循环需要能有效地停止

5.2 用for x语句实现循环结构

for x 语句的一般形式

for x 语句的形式

- for x　　语句
  其它语句
- 当表达式x的计算结果为true时，执行循环语句，否则执行后面的其它语句。
- x是用来控制循环是否执行，称为循环条件。循环时执行的语句可以是一个语句，也可以是多条语句构成的代码块，称为循环体。

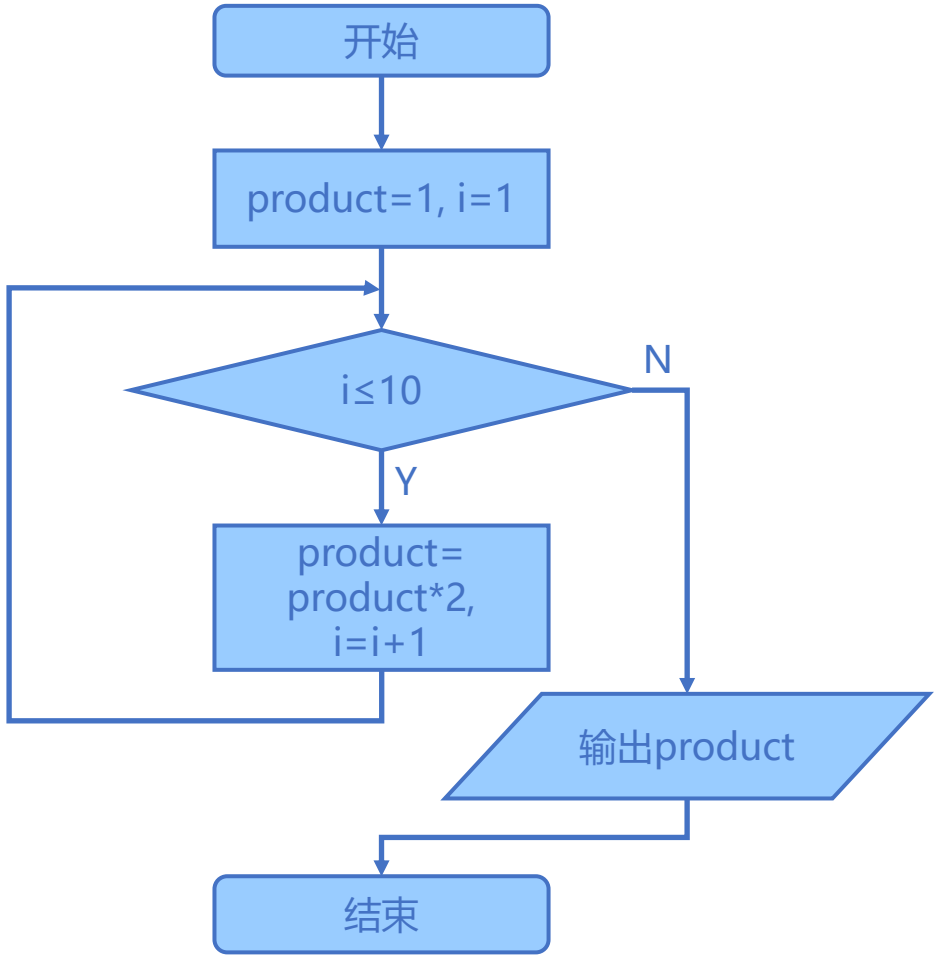


【例5.1】求2^{10}，即2×2×2×2×2×2×2×2×2×2的结果。

5.2用for x 语句实现循环结构

for x 语句的一般形式

【例5.1】求2^{10}，即2×2×2×2×2×2×2×2×2×2的结果。



	A	B	C
1	1	1	/初始化product和i
2	for B1<=10	>A1=A1*2	/当i≤10时循环计算product
3		>B1=B1+1	/循环次数i+1
4	>output("product="/A1)		/输出结果

结果：

```
product=1024
```

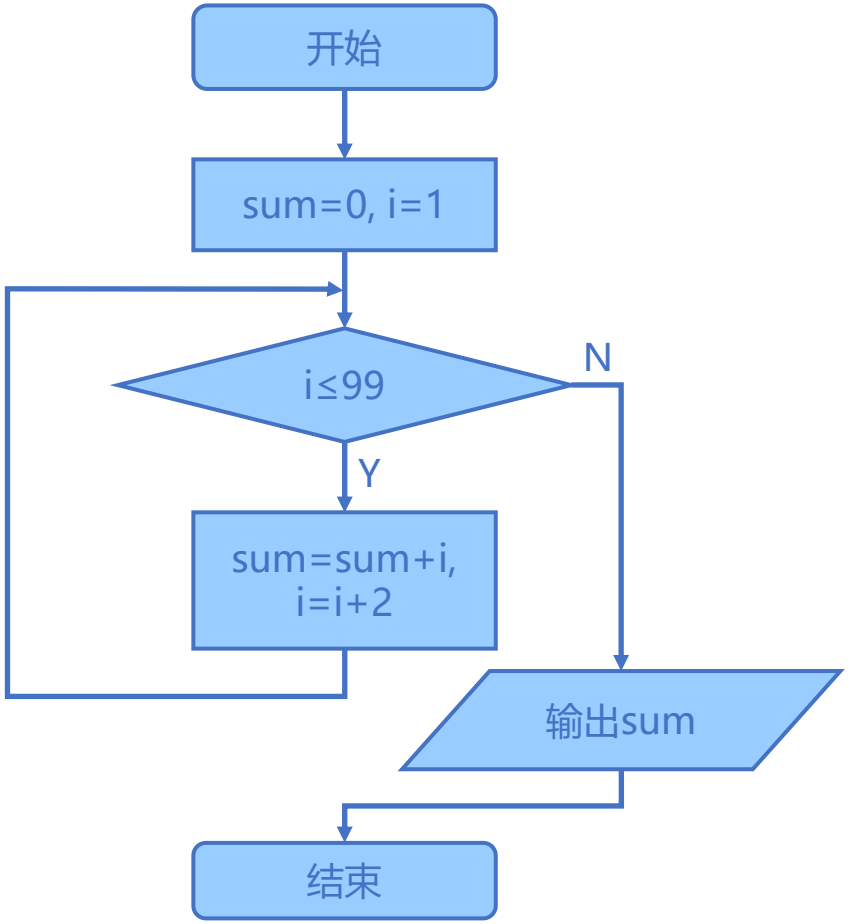
注意

- 循环体如果包含一个以上的语句，应该置于for语句的**代码块**中。
- 要注意给 product 和 i 赋初值。
- 每次执行循环后，要改变 i 的值，使得循环条件改变，让循环能够正常结束。

【例5.2】计算1+3+5+7+...+97+99的结果。

5.2用for x 语句实现循环结构

【例5.2】计算1+3+5+7+...+97+99的结果。



	A	B	C
1	0	1	/初始化sum和i
2	for B1<=99	>A1=A1+B1	/当i≤99时循环计算sum
3		>B1=B1+2	/加数项i+2
4	>output("sum="/A1)		/输出结果

结果：

```
sum=2500
```

for循环的执行过程

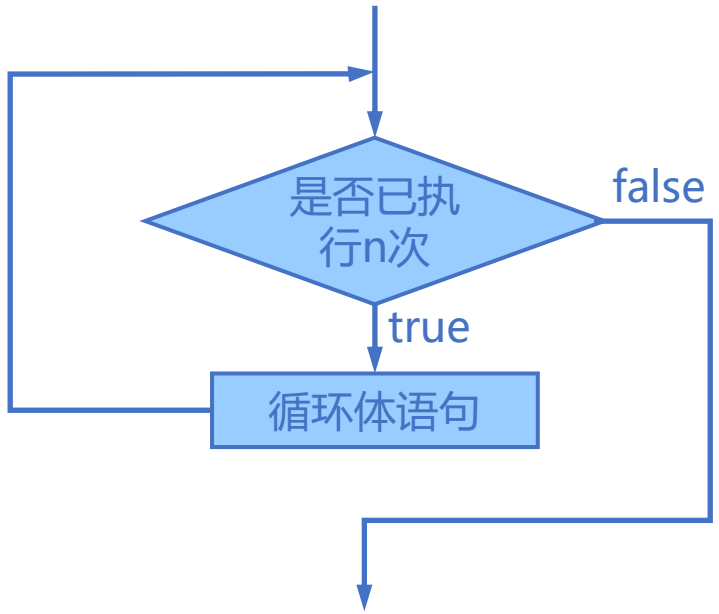
- 先计算循环条件中的表达式，当结果为true时执行循环体，否则将跳过整个循环体，继续执行循环结构后面的语句。
- 循环体可以由一条语句构成，也可以是多条语句构成的代码块。循环体中的语句执行后，会返回for单元格，重新计算循环条件中的表达式，再次判断是否执行循环。
- 在循环体中，需要执行使得循环趋于结束的语句，如改变循环变量，否则循环会重复执行永不结束，称为死循环。

5.2用for x 语句实现循环结构

指定次数的循环

for n 语句的形式

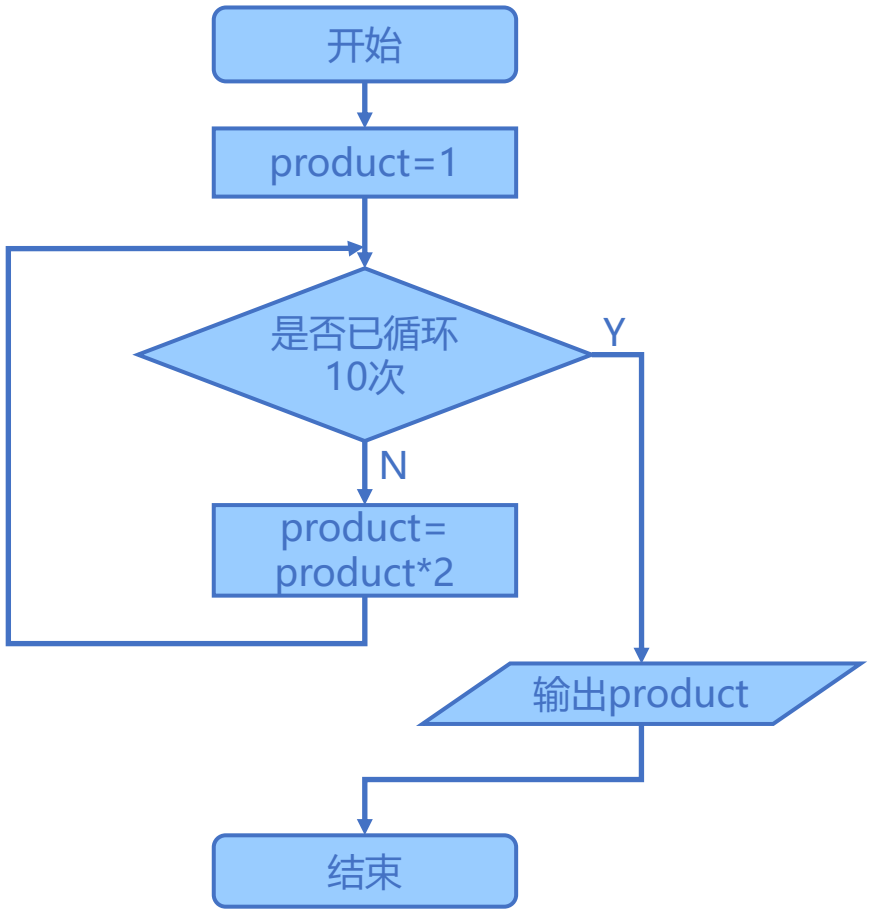
- for n　　循环体语句
  其它语句
- 循环体中的语句将被循环执行n次，执行完毕后执行循环结构后面的其它语句。
- n是正整数，用来指定循环次数。循环执行时，循环主格的值依次为从1至n。



【例5.3】用指定次数的循环结构，计算例5.1：求2^{10}，即2×2×2×2×2×2×2×2×2×2的结果。

5.2用for x 语句实现循环结构

【例5.3】用指定次数的循环结构，计算例5.1：求2^{10}，即2×2×2×2×2×2×2×2×2×2的结果。



	A	B	C
1	1		/初始化product
2	for 10	>A1=A1*2	/当i≤10时循环计算product
3	>output("product="/A1)		/输出结果

结果：

product=1024

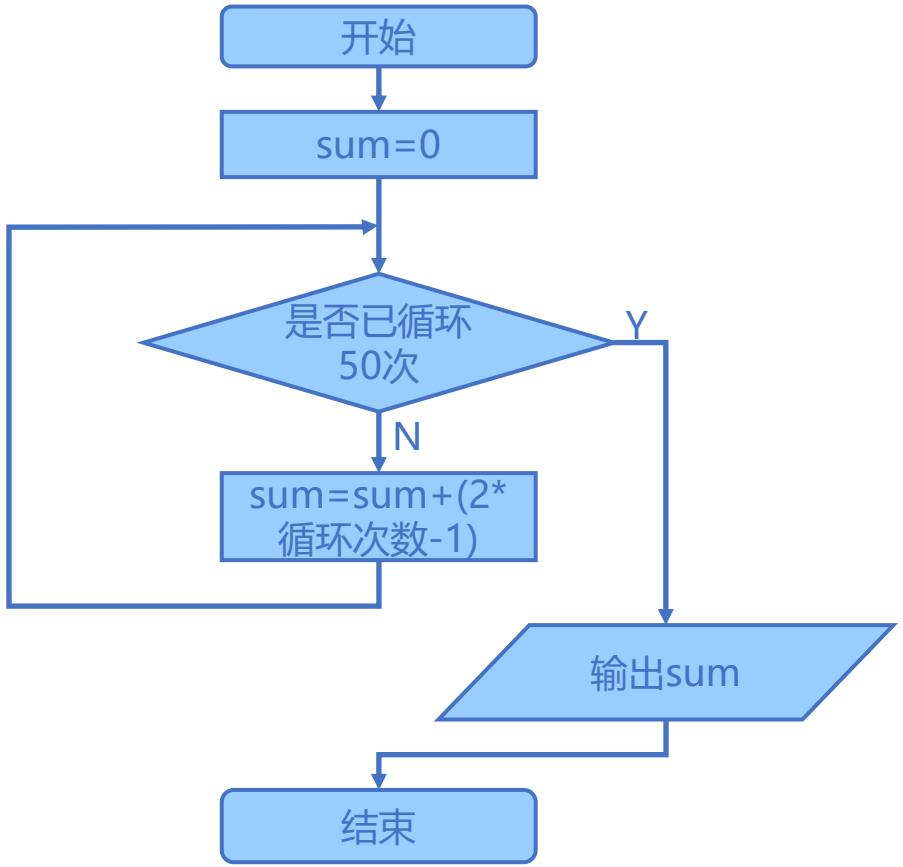
【例5.4】用指定次数的循环结构，计算例5.2：计算1+3+5+7+...+97+99的结果。

5.2用for x 语句实现循环结构

【例5.4】用指定次数的循环结构，计算例5.2：计算1+3+5+7+...+97+99的结果。

解题思路

题目需要计算前50个奇数的和，每一次用来相加的奇数可以根据循环次数计算。



	A	B	C
1	0		/初始化sum
2	for 50	>A1=A1+(2*A2-1)	/循环计算sum
3	>output("sum="/A1)		/输出结果

结果：

```
sum=2500
```

注意

- 在指定次数的循环执行时，循环主格的值为每次循环的次数。
- 指定次数的循环结束后，循环主格的值为循环总次数+1。

改变循环的状态

break

提前终止循环

跳到循环体外，直接执行循环结构后面的语句

break语句只能用于for循环的代码块中

next

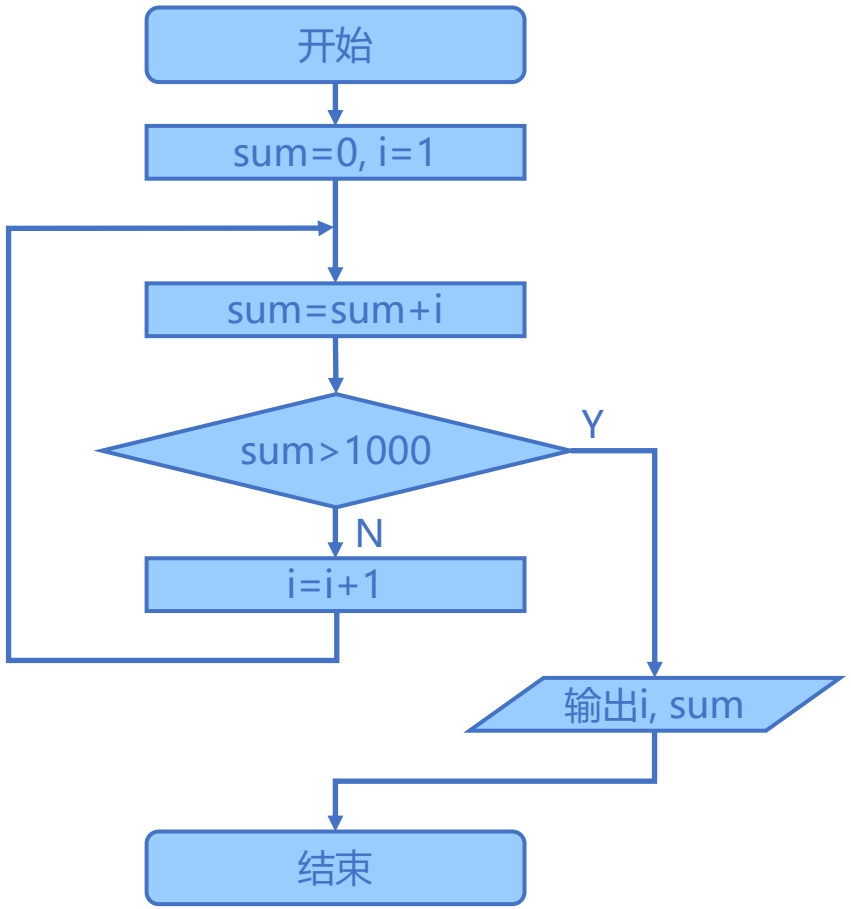
提前结束本次循环

跳过循环体中剩余的语句，返回循环判断语句，判断是否执行下一次循环

next语句只能用于for循环的代码块中

【例5.5】计算正整数连加1+2+3+4+……，请问在计算到第几个数时得到的和刚好超过1000。

【例5.5】计算正整数连加1+2+3+4+……，请问在计算到第几个数时得到的和刚好超过1000。



	A	B	C	D
1	0	1		/初始化sum和i
2	for	>A1=A1+B1		/循环，计算sum
3		if A1>1000	break	/sum刚好超过1000，结束
4		>B1=B1+1		/加数i+1
5	>output("Add to "/B1/". The sum is "/A1/".")			/输出i和sum

结果：

```
Add to 45. The sum is 1035.
```

注意

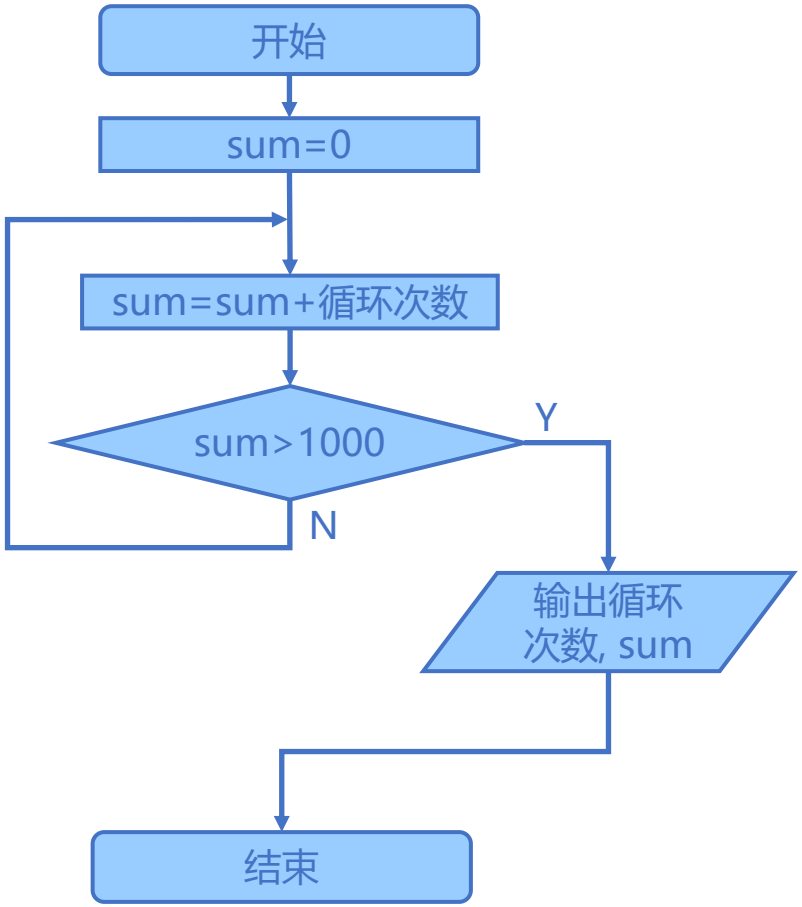
- 在for循环中不添加条件时，循环会一直执行，称为死循环。
- 执行死循环时，必须在循环体中用break语句保证循环有效终止。

【例5.5】计算正整数连加1+2+3+4+……，请问在计算到第几个数时得到的和刚好超过1000。

即使是用for执行死循环，同样可以在循环主格中获得每次的循环次数，代码可以据此简化。

【例5.5】计算正整数连加1+2+3+4+……，请问在计算到第几个数时得到的和刚好超过1000。

即使是用for执行死循环，同样可以在循环主格中获得每次的循环次数，代码可以据此简化。



	A	B	C	D
1	0			/初始化sum
2	for	>A1=A1+A2		/循环，计算sum
3		if A1>1000	break	/sum刚好超过1000，结束
4	>output("Add to "/A2/". The sum is "/A1/".")			/输出i和sum

结果：

```
Add to 45. The sum is 1035.
```

【例5.6】用next语句，计算1+3+5+7+…+97+99的结果。

解题思路

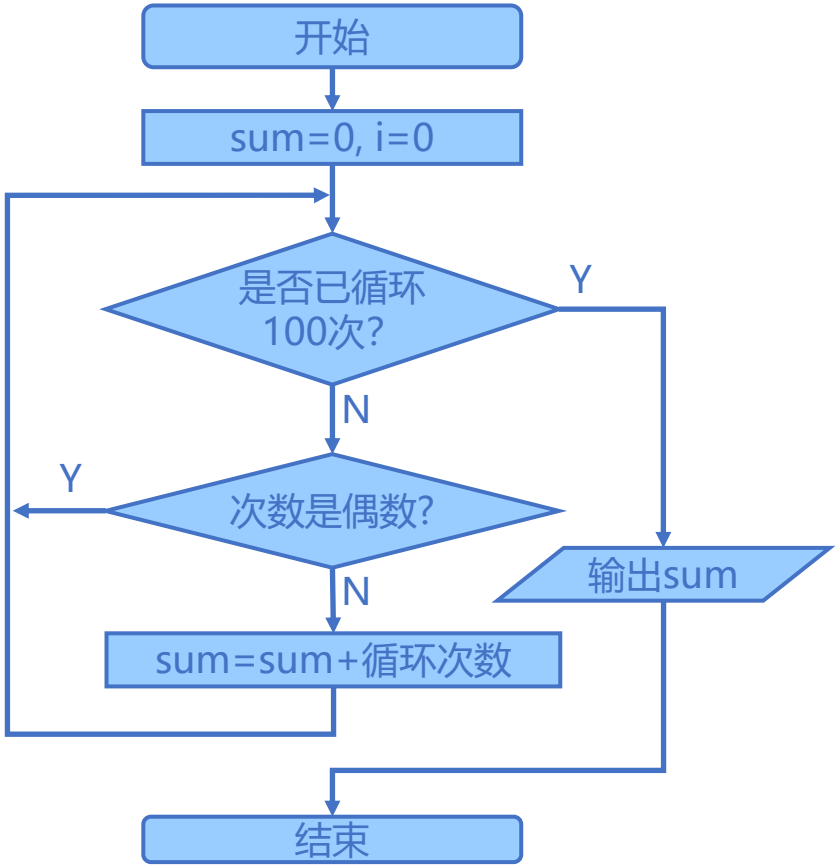
循环100次，用循环次数直接用作加数i。当i为偶数时，跳过当前循环，直接取下一个数判断。当i为奇数时，累加到sum求和。

5.2用for x 语句实现循环结构

【例5.6】用next语句，计算1+3+5+7+…+97+99的结果。

解题思路

循环100次，用循环次数直接用作加数i。当i为偶数时，跳过当前循环，直接取下一个数判断。当i为奇数时，累加到sum求和。



	A	B	C	D
1	0			/初始化sum
2	for 100	if A2%2==0	next	/i是偶数则执行下一次循环
3		>A1=A1+A2		/i是奇数，累加到sum
4	>output("The sum is "/A1/".")			/输出sum

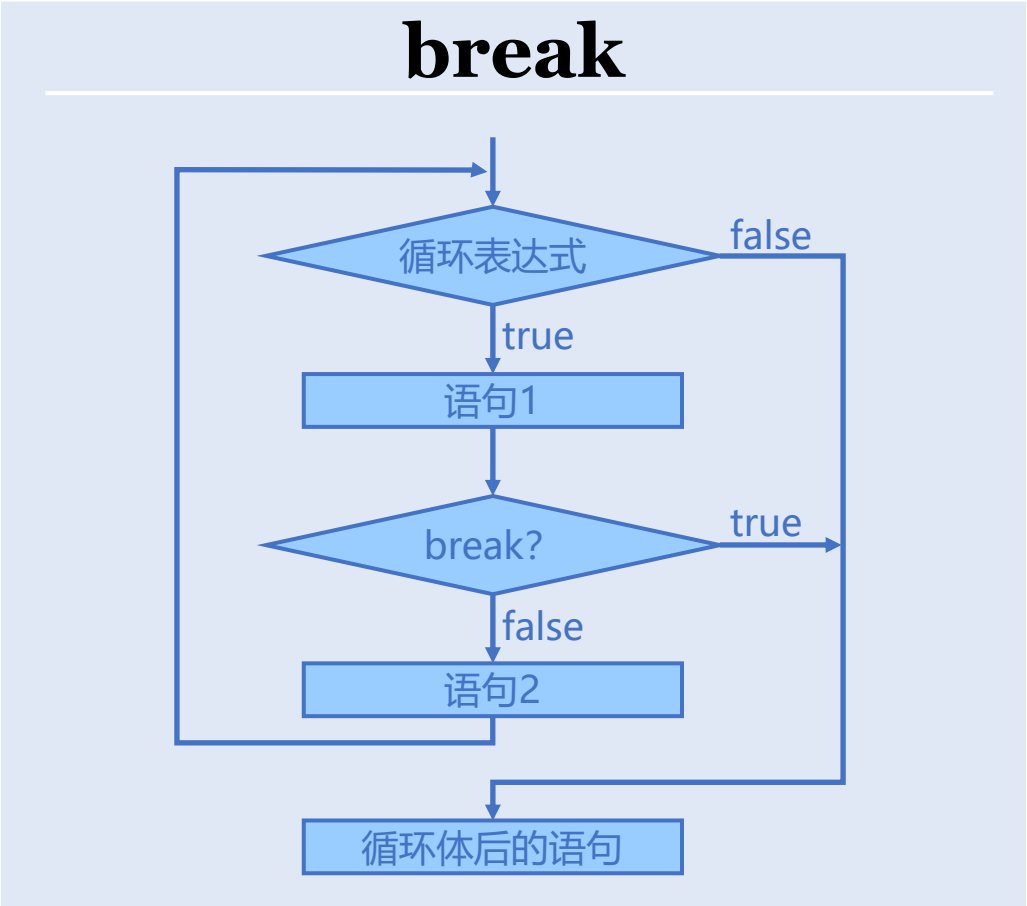
结果：

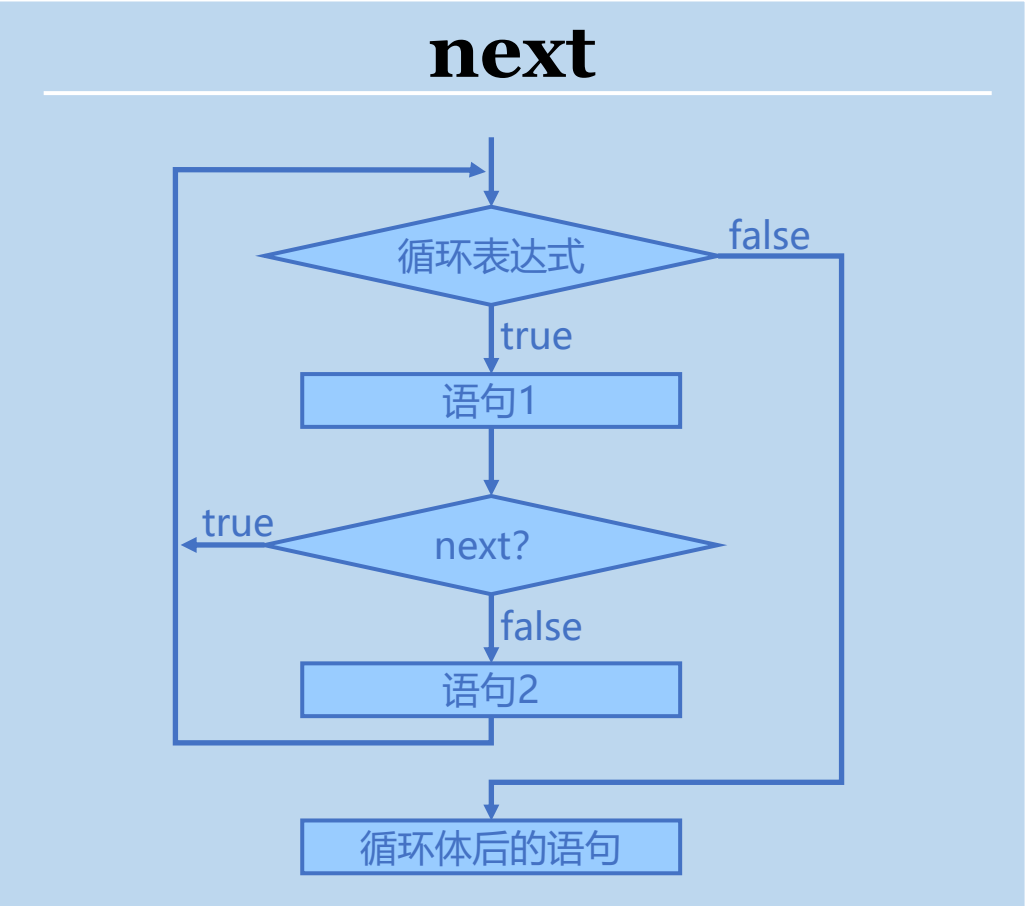
```
The sum is 2500.
```

注意

- next语句用来终止本次循环，返回循环主格执行下一次循环。
  break语句用来终止全部循环，直接退出循环执行后面的语句。

break和next语句的比较

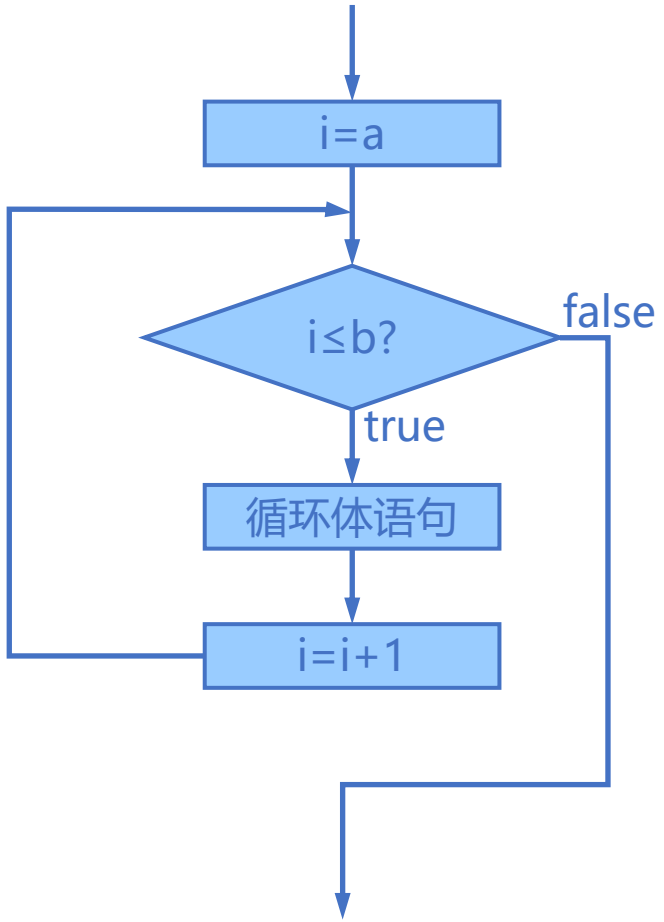




5.3 指定循环范围的for语句

for a,b 语句的形式

- for a,b　　循环体语句
  其它语句
- 执行循环时，循环主格的值依次为从a至b，每次加1。
- 当循环主格的值递增到大于b时，循环执行完毕，将执行循环结构后面的其它语句。



【例5.7】找出所有各位数字和为10的三位数。

解题思路

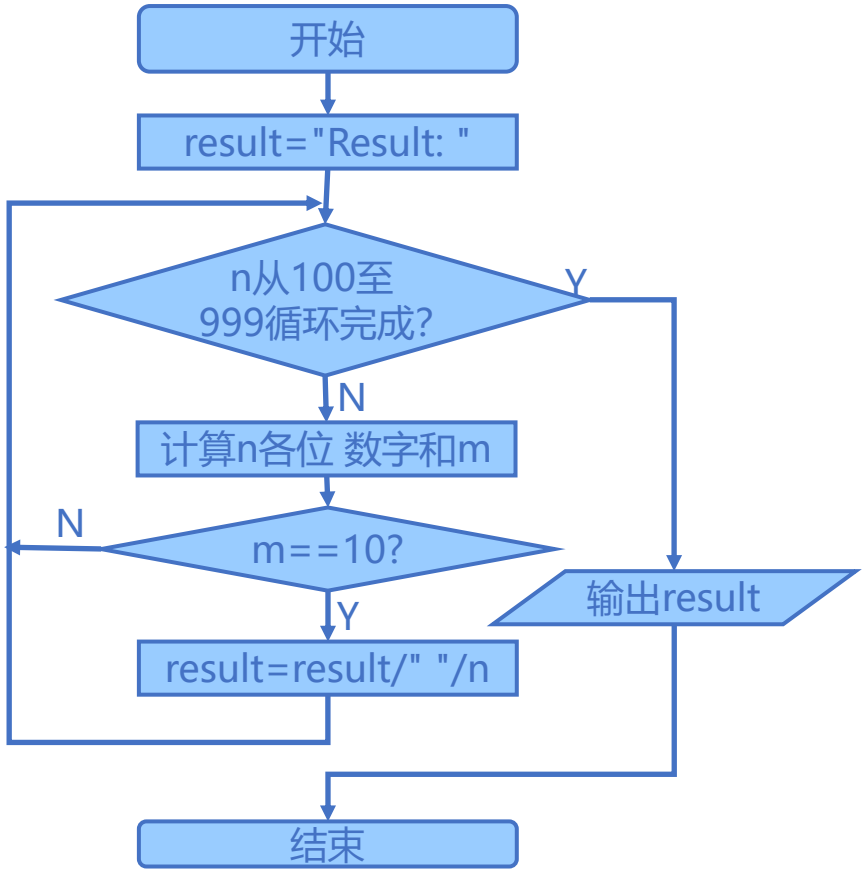
从100至999循环所有三位数，判断各位数字的和是否为10，将符合条件的三位数记录下来，在循环完成后输出。

5.3指定循环范围的 for 语句

【例5.7】找出所有各位数字和为10的三位数。

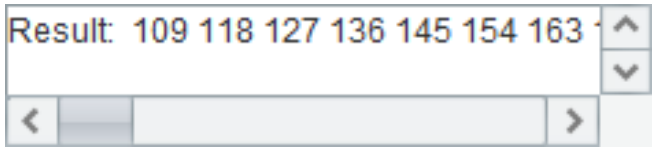
解题思路

从100至999循环所有三位数，判断各位数字的和是否为10，将符合条件的三位数记录下来，在循环完成后输出。



	A	B	C	D
1	Result:			/用A1记录输出结果
2	for 100,999	=A2%10+(A2\10)%10+A2\100		/计算n各位数字的和m
3		if B2==10	>A1=A1/" "/A2	/如果m为10，记录当前数
4	>output(A1)			/输出A1作为结果

结果：



注意

- \运算符在操作数均为整数时，为整除计算符。

5.3指定循环范围的 for 语句

for a,b,s 语句的形式

- for a,b,s　　循环体语句
  其它语句
- 执行循环时，循环主格的值依次为从a至b，每次加s。
- 当循环主格的值递增到大于b时，循环执行完毕，将执行循环结构后面的其它语句。

【例5.8】用公式$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots\right)$的前500,000项计算π的近似值。

解题思路

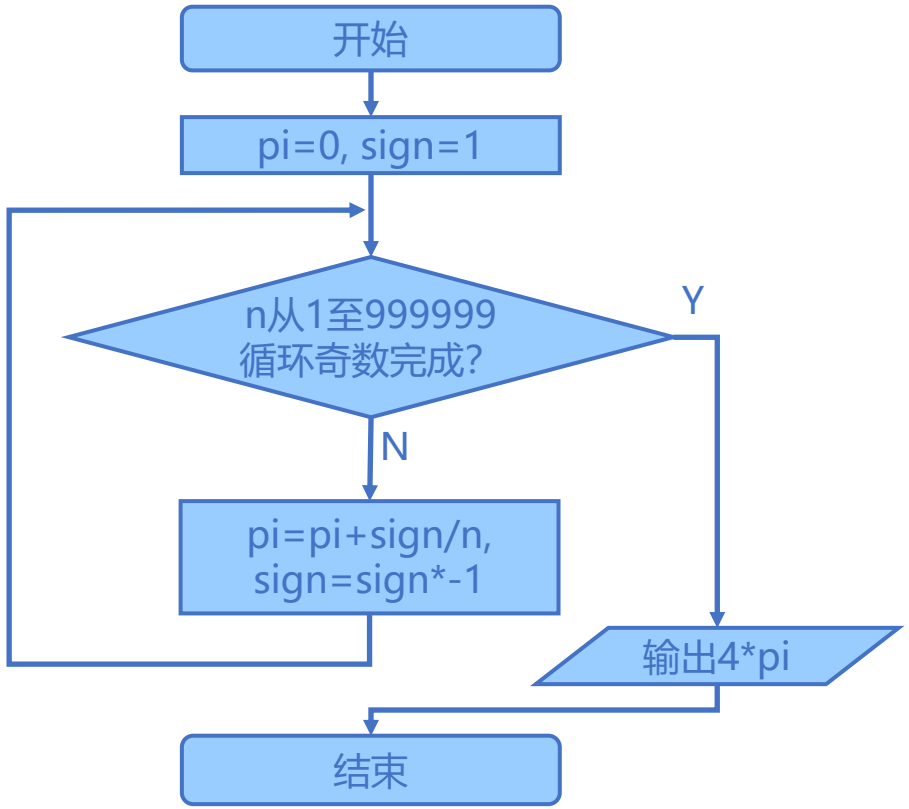
公式中每一项的正负号轮换，分母分别为1,3,5,...第500,000项分母为999,999.

5.3指定循环范围的 for 语句

【例5.8】用公式$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots\right)$的前500,000项计算π的近似值。

解题思路

公式中每一项的正负号轮换，分母分别为1,3,5,...第500,000项分母为999,999.



	A	B	C
1	0	1	/初始化pi,sign
2	for 1,999999,2	>A1=A1+B1/A2	/用公式累加项
3		>B1=B1*(-1)	/变换正负号
4	>output("π≈"/(A1*4))		/输出π的近似值

结果：

π≈3.141590653589692

注意

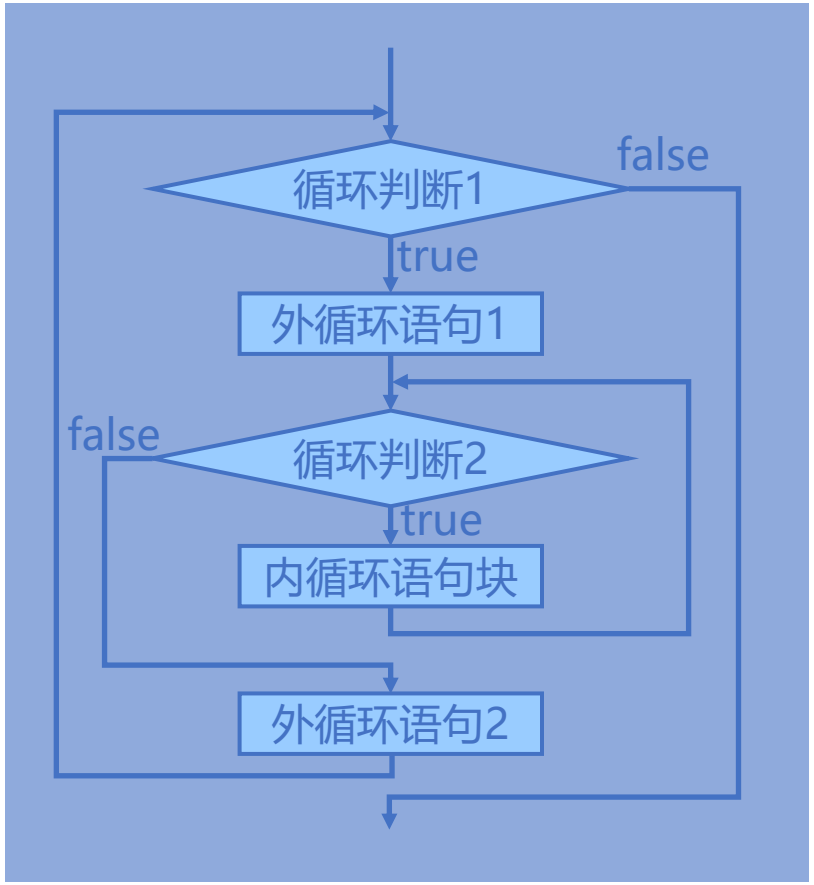
- for a,b,s将在循环值大于b时结束，因此A2中如果是for 1,1000000,2, 执行循环的效果是相同的。

5.4 循环结构的嵌套

5.4 循环结构的嵌套

	A	B	C	D
1	for …	外循环语句1		/外层循环语句
2		for …	内循环语句1	/内层循环语句
3			内循环语句2	/内层循环语句
4		外循环语句2		/外层循环语句
5	其它语句			/循环体外语句

- 循环嵌套也称为多重循环，在外层循环的每一次执行中，都需执行整个内层循环。
- 当外层循环全部执行完毕后，才执行循环结构后面的语句。



5.4 循环结构的嵌套

【例5.9】输出斐波那契(Fibonacci)数列，每6项一行，输出5行。斐波那契数列的前两项均为1，从第3项起，均为前两项的和，即为1,1,2,3,5,8,…

解题思路

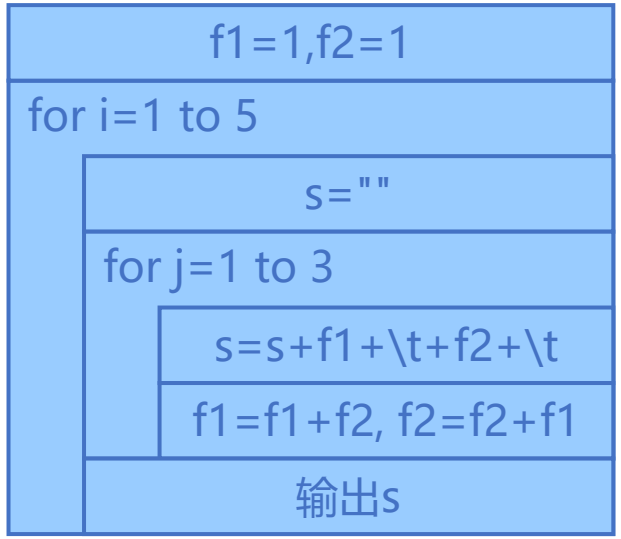
外层循环每次输出一行。内层每行循环3次，每次记录两项，并计算下两项的值。

5.4 循环结构的嵌套

【例5.9】输出斐波那契(Fibonacci)数列，每6项一行，输出5行。斐波那契数列的前两项均为1，从第3项起，均为前两项的和，即为1,1,2,3,5,8,...

解题思路

外层循环每次输出一行。内层每行循环3次，每次记录两项，并计算下两项的值。



	A	B	C	D	E
1	1	1			/初始化数列前两项
2	for 5	=""			/初始化输出串
3		for 3	>B2=B2/A1/"\t"/B1/"\t"		/向输出串中添加项
4			>A1=A1+B1	>B1=B1+A1	/计算下面两项
5		>output(B2)			/输出一行

结果：

1	1	2	3	5	8
13	21	34	55	89	144
233	377	610	987	1597	2584
4181	6765	10946	17711	28657	46368
75025	121393	196418	317811	514229	832040

在多层循环中使用break和next

- 在使用break和next时，都可以指定循环主格C。通过使用break C和next C，指明终止或进入下次循环的是哪一层。
- 当省略循环主格C时，break和next语句所控制的是语句格所在的最内层的循环。

【例5.10】输出大于200的前50个质数。

解题思路

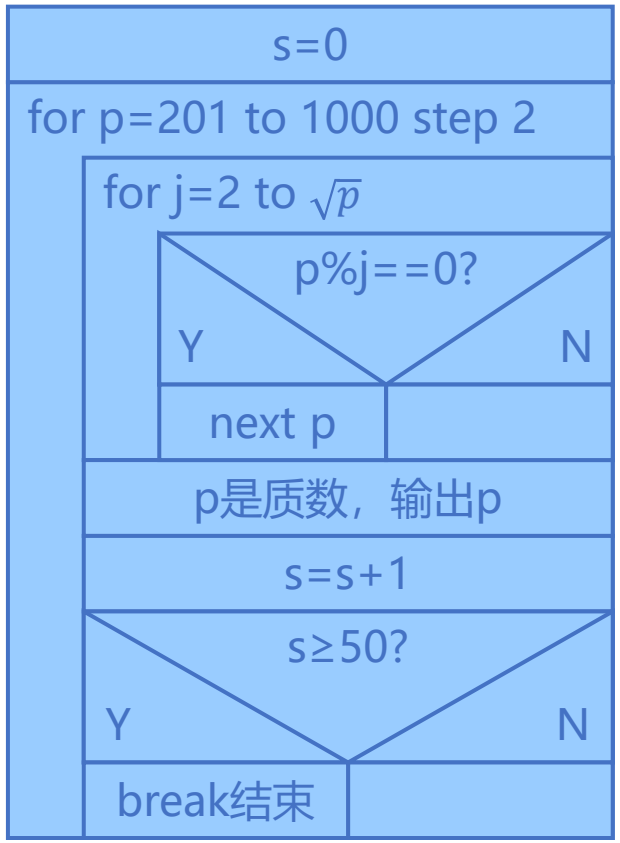
质数 p 需要满足：它无法被从2到 $\sqrt{p}$ 的所有整数整除。从201开始判断各个奇数是否是质数。

在多层循环中使用break和next

【例5.10】输出大于200的前50个质数。

解题思路

质数 p 需要满足：它无法被从2到 $\sqrt{p}$ 的所有整数整除。从201开始判断各个奇数是否是质数。



	A	B	C	D	E
1	0				/A1记录质数个数
2	for 201,1000,2	=sqrt(A2)			/循环，计算A2方根
3		for 2,B2	if A2%B3==0	next A2	/合数，判断下一个
4		>output(A2)	>A1=A1+1		/质数，输出并计数
5		if A1>=50	break		/输出完成，结束

结果：

5.5 循环结构举例

【例5.11】给出两个正整数，求它们的最大公约数和最小公倍数。

解题思路

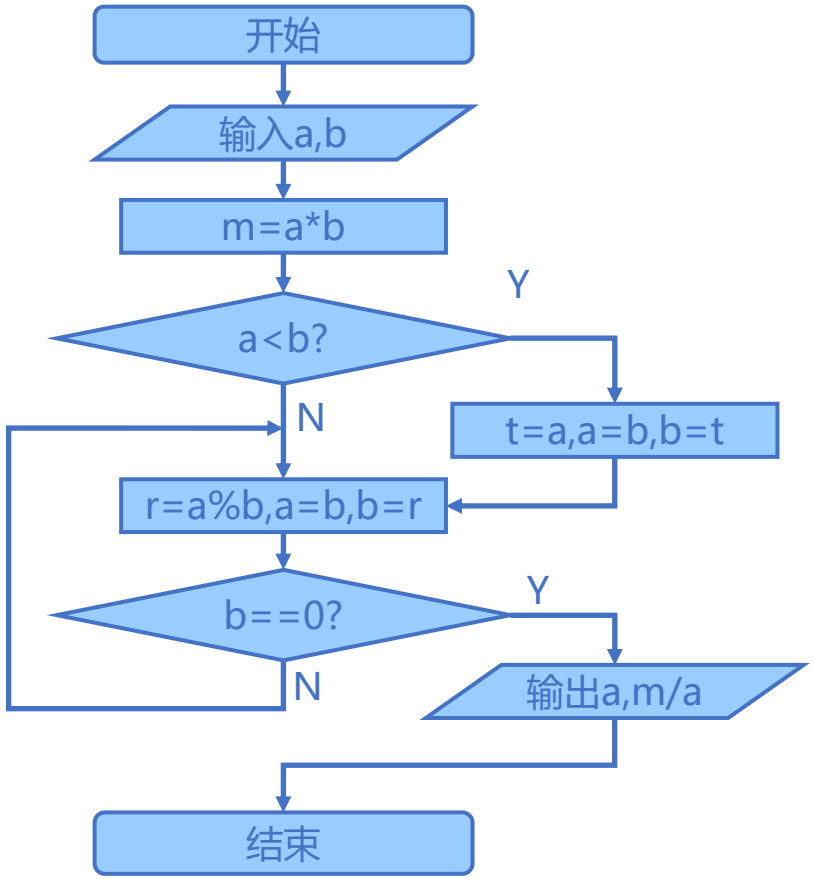
使用辗转相除法，两个正整数的最大公约数等于它们相除的余数，与较小数的最大公约数。循环相除，直到较小数是较大数的约数为止。同时，最小公倍数和最大公约数相乘等于两数的乘积。

5.5 循环结构举例

【例5.11】给出两个正整数，求它们的最大公约数和最小公倍数。

解题思路

使用辗转相除法，两个正整数的最大公约数等于它们相除的余数，与较小数的最大公约数。循环相除，直到较小数是较大数的约数为止。同时，最小公倍数和最大公约数相乘等于两数的乘积。



	A	B	C	D
1	144	96	=A1*B1	/输入两个数，计算乘积
2	if A1<B1	>(t=A1,A1=B1,B1=t)		/如果a<b，交换两个数
3	for B1!=0	>(r=A1%B1,A1=B1,B1=r)		/辗转相除
4	>output("GCD:"/A1/", LCM:"/(C1\A1))			/输出GCD和LCM

结果：

```
GCD:48, LCM:288
```

注意

- 括号运算符: $(x_1, x_1, \cdots)$ 可以依次计算表达式$x_1, x_1, \cdots$，能够有效简化代码，整数除法用\符号。

5.5 循环结构举例

【例5.12】井中一只青蛙向上爬，白天爬5尺，夜里滑下2尺，井高20尺，求几天能爬出。

解题思路

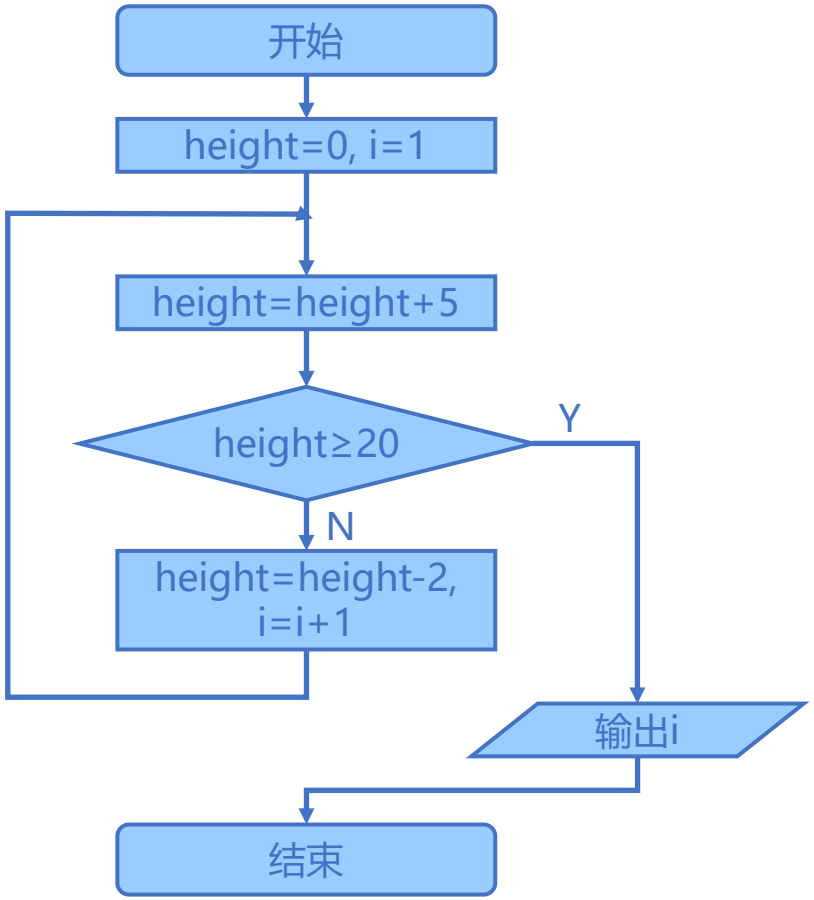
循环，只需每天白天爬上后判断是否已经成功爬出。

5.5 循环结构举例

【例5.12】井中一只青蛙向上爬，白天爬5尺，夜里滑下2尺，井高20尺，求几天能爬出。

解题思路

循环，只需每天白天爬上后判断是否已经成功爬出。



	A	B	C	D
1	0	1		/初始化height和i
2	for	>A1=A1+5		/当未爬出井时始终循环
3		if A1>=20	break	/白天爬出井口，终止循环
4		>A1=A1-2	>B1=B1+1	/夜里滑下，天数i+1
5	>output("Cost "/B1/" days.")			/输出天数

结果：

```
Cost 6 days.
```

【例5.13】找出所有型为AABB的四位平方数。

解题思路

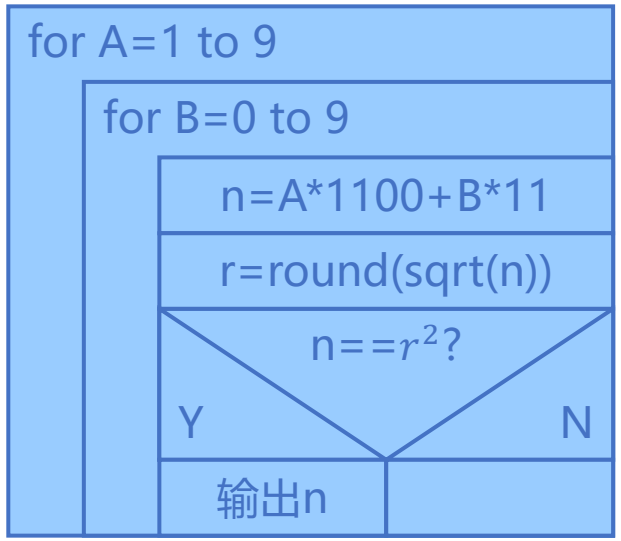
A循环1~9，B循环0~9，组合成AABB的四位数，再判断它是不是平方数。判断时由于双精度数计算时可能存在误差，因此需要用round函数四舍五入计算平方根，再看其平方是否等于原数。

5.5 循环结构举例

【例5.13】找出所有型为AABB的四位平方数。

解题思路

A循环1~9，B循环0~9，组合成AABB的四位数，再判断它是不是平方数。判断时由于双精度数计算时可能存在误差，因此需要用round函数四舍五入计算平方根，再看其平方是否等于原数。



	A	B	C	D	E
1	for 9	for 0,9	=A1*1100+B1*11		/循环所有可能的ABBA
2			=round(sqrt(C1))		/四舍五入计算平方根
3			if C2*C2==C1	>output(C1)	/如果ABBA是平方数，输出

结果：

```
7744
```

5.5 循环结构举例

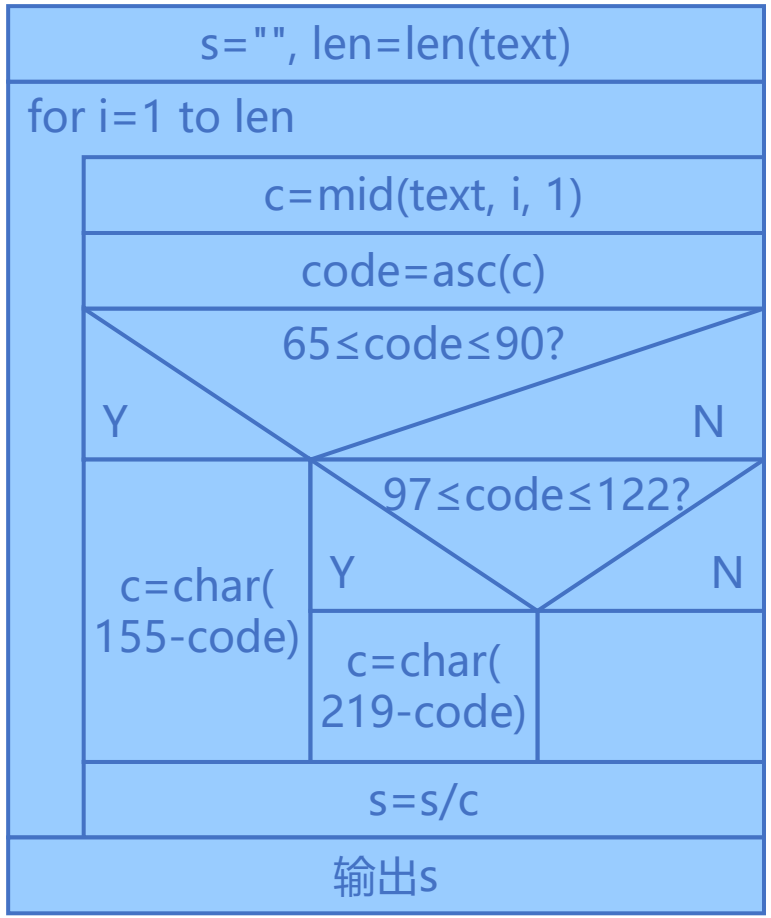
【例5.14】将一段英文作为参数输入，对其执行简单加密转换：将a转换为z，b转换为y，…，z转换为a，A转换为Z，B转换为Y，…，Z转换为A，其它字符不变。输出加密后的文字。（需要翻译时可以将加密后的文字作为参数输入，再次执行转换。）

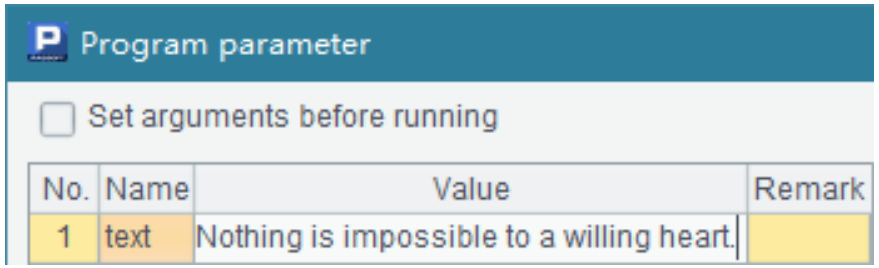
解题思路

大写字符的ASCII码从A到Z为65~90，小写字母的ASCII码从a到z为97~122。这样，转码后，小写字母ASCII码为90+65-n，大写字母ASCII码为97+122-n。

5.5 循环结构举例

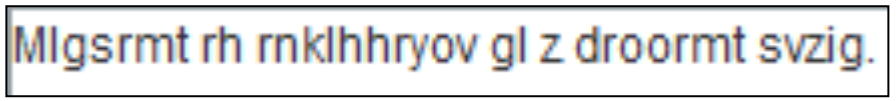
【例5.14】将一段英文作为参数输入，对其执行简单加密转换：将a转换为z，b转换为y，…，z转换为a，A转换为Z，B转换为Y，…，Z转换为A，其它字符不变。输出加密后的文字。（需要翻译时可以将加密后的文字作为参数输入，再次执行转换。）





	A	B	C	D
1		=len(text)		/初始化加密串和字符数
2	for B1	=mid(text, A2, 1)	=asc(B2)	/取单个字符，计算ASCII
3		if C2>=65&&C2<=90	>B2=char(155-C2)	/大写字母加密
4		else if C2>=97&&C2<=122	>B2=char(219-C2)	/小写字母加密
5		>A1=A1/B2		/字符添加到结果
6	>output(A1)			/输出加密串

结果：



5.6 习题

5.6 习题

【习题5.1】输入一个正整数n（如10），计算它的阶乘n!。

【习题5.2】找出所有的3位水仙花数：它所有位上数字的立方和，等于它本身。

【习题5.3】用迭代法计算1234的平方根，迭代法求 $x=\sqrt{a}$ 的迭代公式是：$x_{n+1}=\frac{1}{2}\left(x_n+\frac{a}{x_n}\right)$，要求迭代到前后两项误差小于$1^{-10}$。

【习题5.4】明代程大位的著作《算法统宗》中有一道“百僧百馍”问题：一百馒头一百僧，大僧三个更无增，小僧三人分一个，大小和尚各几丁。请输出所有可能的大小和尚数目。

【习题5.5】一起盗窃案审理中有四个嫌疑人甲乙丙丁。甲说“不是我，是乙偷的。”乙说：“罪犯在丙丁之中。”丙说：“甲说的是对的。”丁说：“罪犯在他们三个之中。”如果4个人中有2个说了真话，2个说了假话，而且罪犯只有1名，那么，罪犯是谁呢?

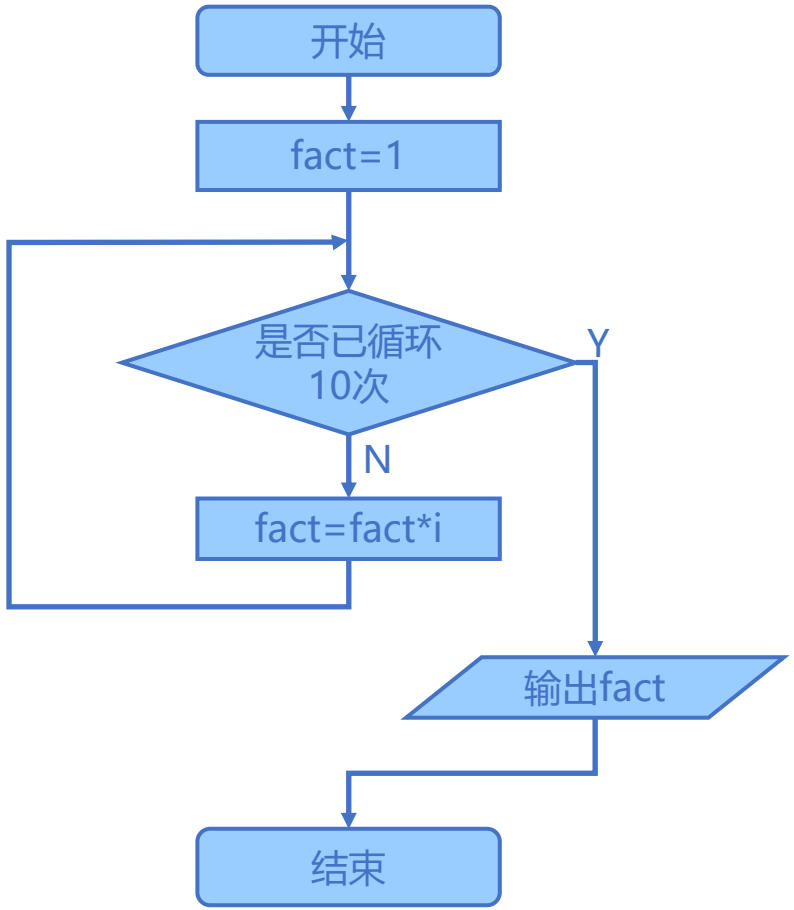
【习题5.6】输入一段文字，统计其中英文字符与非英文字符各有多少个。如输入：One man's fault is another man's lesson.

【习题5.7】2020年1月1日是周三，请输出前三个月的月历。在月历中，每周占一行，从周日开始显示。可以用days函数计算某个月的天数，如计算2020年3月的天数，可以用days(date(2020,3,1))。

【习题5.1】输入一个正整数n（如10），计算它的阶乘n!。

5.6 习题

【习题5.1】输入一个正整数n（如10），计算它的阶乘n!。



	A	B	C
1	10	1	/定义n，初始化fact
2	for A1	>B1=B1*A2	/循环计算阶乘
3	>output(A1/"!="/B1)		/输出结果

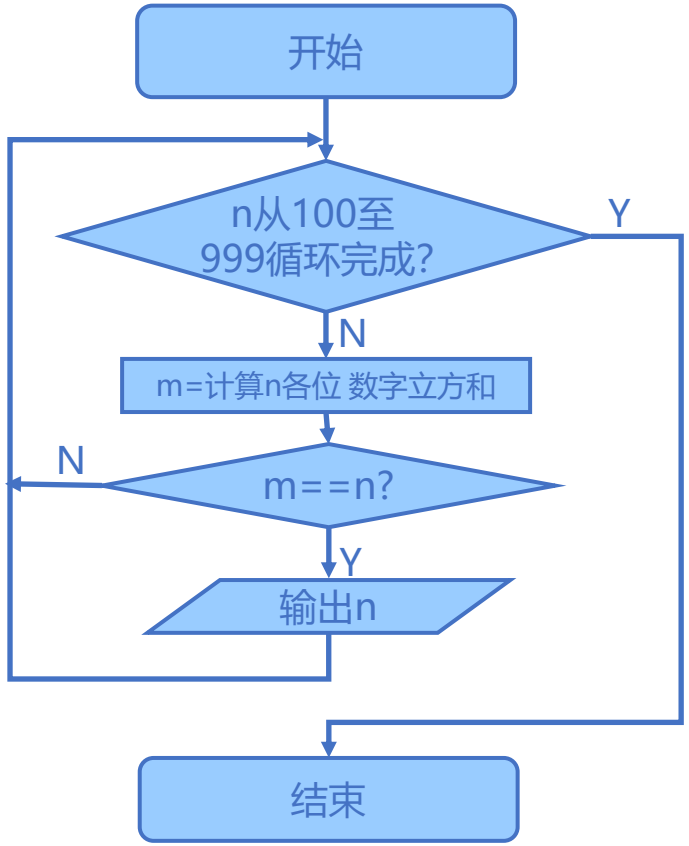
结果：

```
10!=3628800
```

【习题5.2】找出所有的3位水仙花数：它所有位上数字的立方和，等于它本身。

5.6 习题

【习题5.2】找出所有的3位水仙花数：它所有位上数字的立方和，等于它本身。



	A	B	C	D
1	for 100,999	=A1%10	=(A1\10)%10	/计算n各位数字
2		=A1\100	=B1*B1*B1+C1*C1*C1+B2*B2*B2	/计算各位数字立方和m
3		if C2==A1	>output(A1)	/如果是水仙花数，输出

结果：

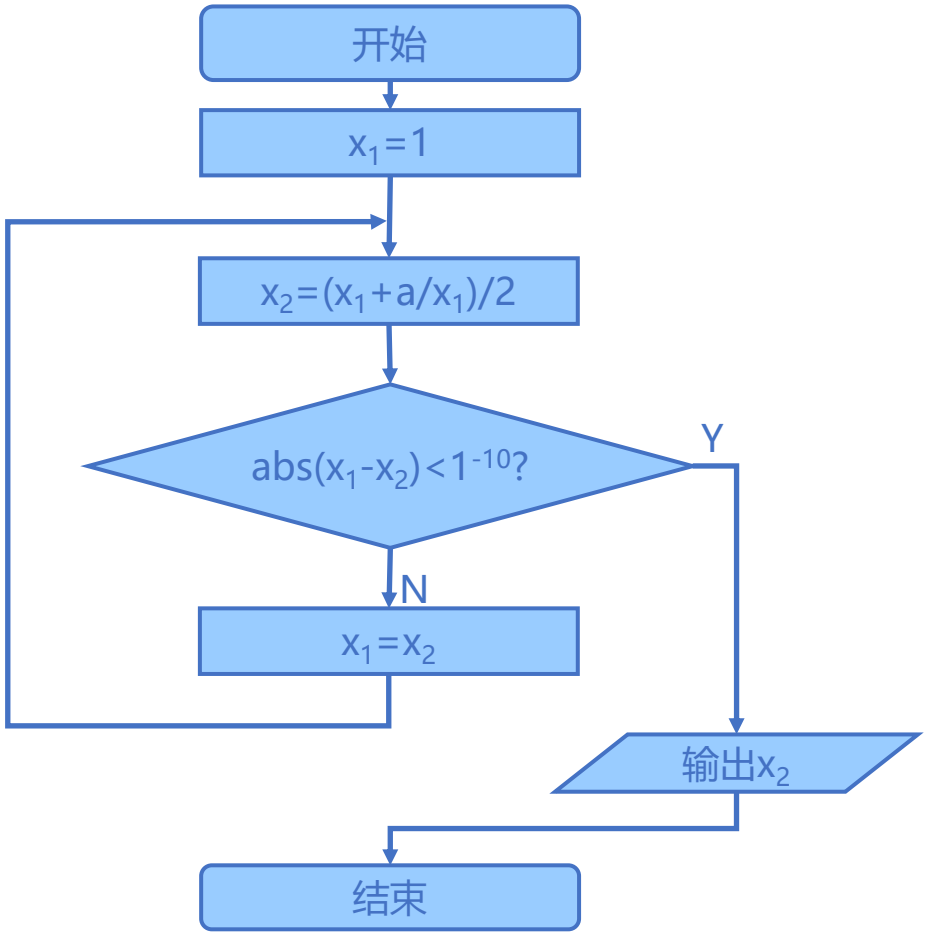
```
153
370
371
407
```

5.6 习题

【习题5.3】用迭代法计算1234的平方根，迭代法求 $x = \sqrt{a}$ 的迭代公式是：$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right)$，要求迭代到前后两项误差小于$1^{-10}$。

5.6 习题

【习题5.3】用迭代法计算1234的平方根，迭代法求 $x=\sqrt{a}$ 的迭代公式是：$x_{n+1}=\frac{1}{2}\left(x_n+\frac{a}{x_n}\right)$，要求迭代到前后两项误差小于$1^{-10}$。



	A	B	C	D
1	1234	1e-10	1	/定义n，精度及初始值
2	for	=(C1+A1/C1)/2		/循环迭代公式
3		if abs(C1-B2)<B1	break	/误差满足条件，循环终止
4		>C1=B2		/更新平方根
5	>output("root="/B2)			/输出结果

结果：

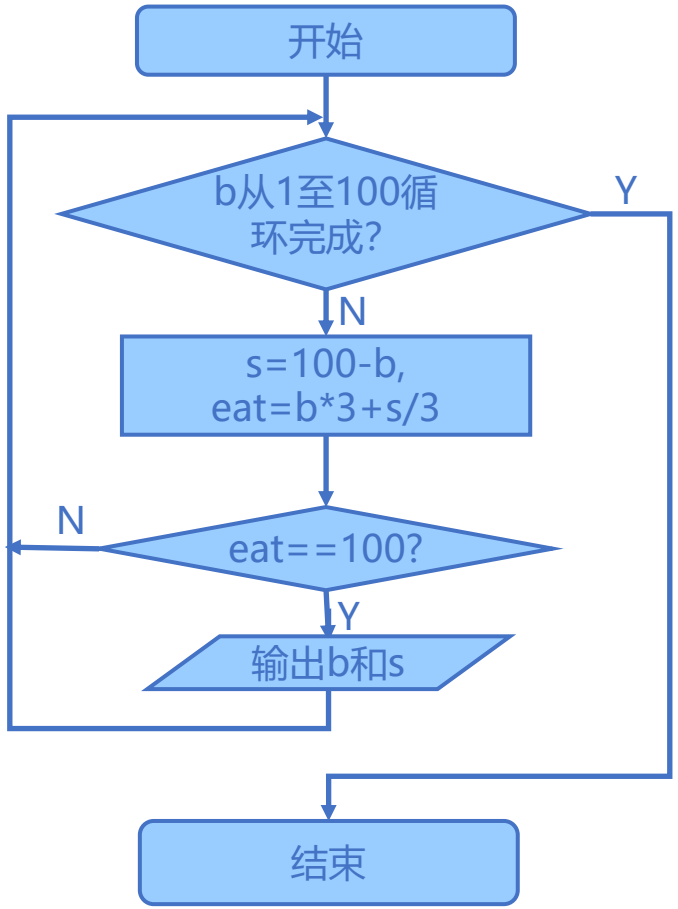
```
root=35.12833614050059
```

5.6 习题

【习题5.4】明代程大位的著作《算法统宗》中有一道“百僧百馍”问题：一百馒头一百僧，大僧三个更无增，小僧三人分一个，大小和尚各几丁。请输出所有可能的大小和尚数目。

5.6 习题

【习题5.4】明代程大位的著作《算法统宗》中有一道“百僧百馍”问题：一百馒头一百僧，大僧三个更无增，小僧三人分一个，大小和尚各几丁。请输出所有可能的大小和尚数目。



	A	B	C	D
1	for 100	=100-A1	=A1*3+B1/3	/循环，计算和尚数和馒头数
2		if C1==100	>output(A1/", "/B1)	/如果满足条件，输出结果

结果：

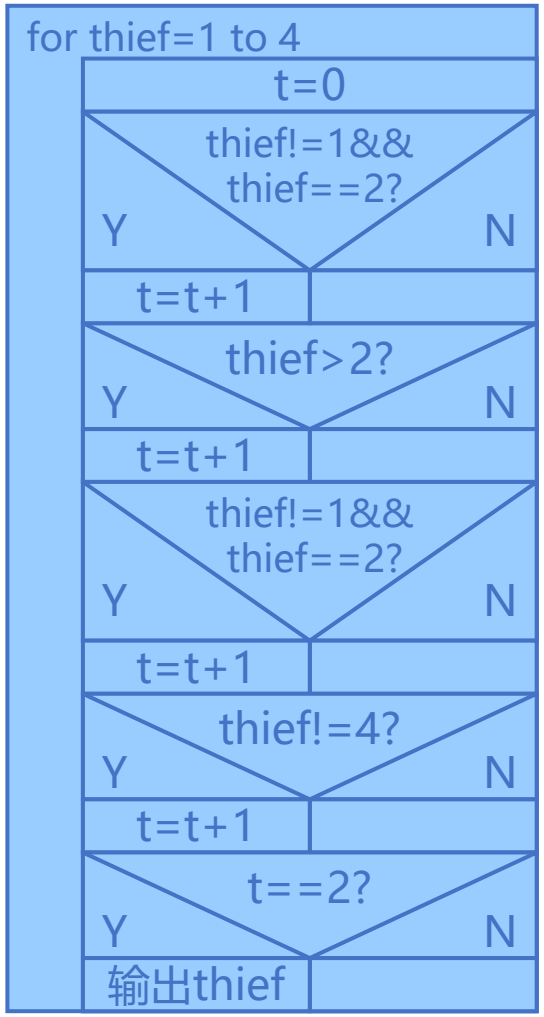
```
25, 75
```

5.6 习题

【习题5.5】一起盗窃案审理中有四个嫌疑人甲乙丙丁。甲说“不是我，是乙偷的。”乙说：“罪犯在丙丁之中。”丙说：“甲说的是对的。”丁说：“罪犯在他们三个之中。”如果4个人中有2个说了真话，2个说了假话，而且罪犯只有1名，那么，罪犯是谁呢?

5.6 习题

【习题5.5】一起盗窃案审理中有四个嫌疑人甲乙丙丁。甲说“不是我，是乙偷的。”乙说：“罪犯在丙丁之中。”丙说：“甲说的是对的。”丁说：“罪犯在他们三个之中。”如果4个人中有2个说了真话，2个说了假话，而且罪犯只有1名，那么，罪犯是谁呢?



	A	B	C	D
1	for 4	=0		/循环罪犯是4个人的情况
2		if A1!=1 && A1==2	>B1=B1+1	/甲是否说了真话
3		if A1>2	>B1=B1+1	/乙是否说了真话
4		if A1!=1 && A1==2	>B1=B1+1	/丙是否说了真话
5		if A1!=4	>B1=B1+1	/丁是否说了真话
6		if B1==2	>output("The thief is "/A1)	/两人说了真话，输出罪犯

结果：

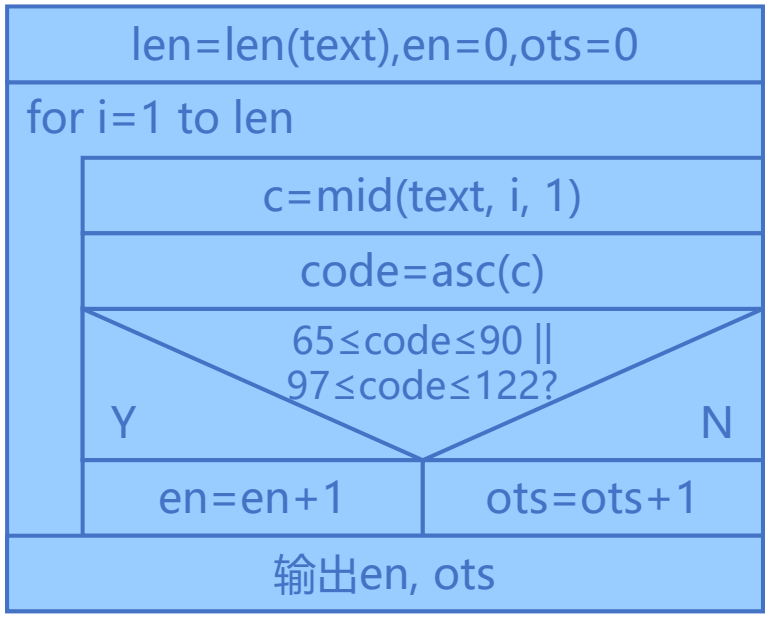
```
The thief is 3
```

5.6 习题

【习题5.6】输入一段文字，统计其中英文字符与非英文字符各有多少个。如输入：
One man's fault is another man's lesson.

5.6 习题

【习题5.6】输入一段文字，统计其中英文字符与非英文字符各有多少个。如输入：
One man's fault is another man's lesson.



Program parameter

Set arguments before running

No.	Name	Value	Remark
1	text	another man's lesson.	

	A	B	C	D
1	=len(text)	0	0	/初始化字符数
2	for A1	=mid(text, A2, 1)	=asc(B2)	/取单个字符，计算ASCII
3		if C2>=65&&C2<=90 || C2>=97&&C2<=122	>B1=B1+1	/英文字符
4		else	>C1=C1+1	/非英文字符
5	>output("En:"/B1/", Ots:"/C1)			/输出结果

结果：

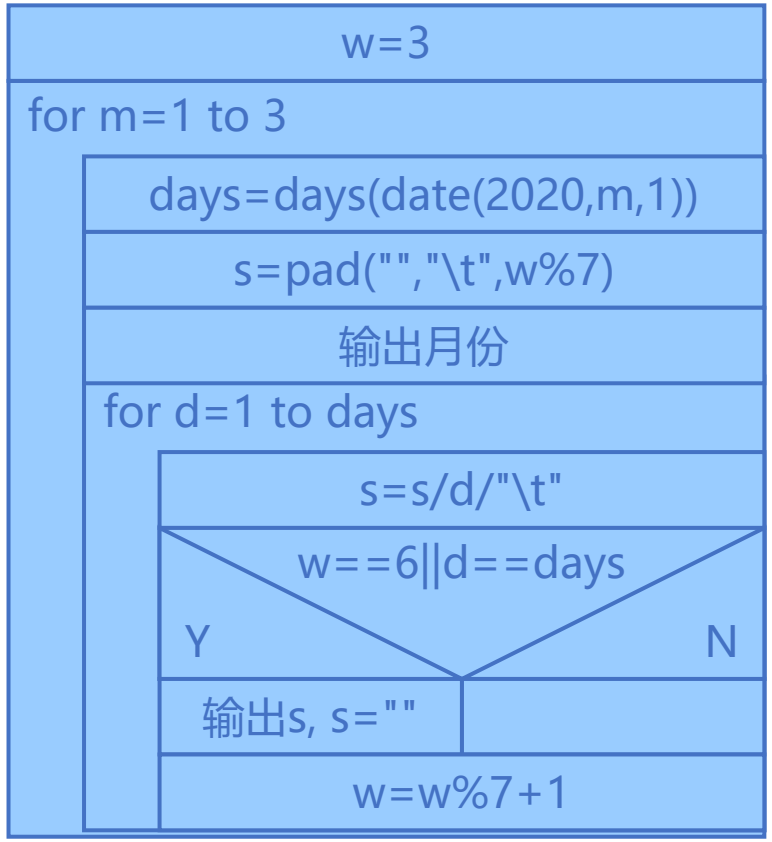
```
En:31, Ots:9
```

5.6 习题

【习题5.7】2020年1月1日是周三，请输出前三个月的月历。在月历中，每周占一行，从周日开始显示。可以用days函数计算某个月的天数，如计算2020年3月的天数，可以用days(date(2020,3,1))。

5.6 习题

【习题5.7】2020年1月1日是周三，请输出前三个月的月历。在月历中，每周占一行，从周日开始显示。可以用days函数计算某个月的天数，如计算2020年3月的天数，可以用days(date(2020,3,1))。



	A	B	C	D	E
1	3				/初始化星期
2	for 3	=date(2020,A2,1)	=days(B2)	=pad("","\t",A1%7)	/本月天数，首周补足空白天数
3		=string(B2,"MMMM")	>output(B3)		/输出月份
4		for C2	>D2=D2/B4/"\t"		/添加本日日期
5			if A1==6 || B4==C2	>output(D2)	/周末或月底，输出
6				>D2=""	/输出后清空字串
7			>A1=A1%7+1		/星期加一

结果：

16	17	18	19	20	21	22
23	24	25	26	27	28	29
March						
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

第6章 文本与字符串

6.1 字符串

- 字符串常数
- 字符串操作符
- 特殊的字符串
- 转义符与特殊字符

6.2 字符串的转换

- 字符与ASCII码
- 编码转换
- 字符串与显示格式

6.3 动态程序

- 宏
- eval函数

6.4 文本文件

6.5 字符串函数

- 字符函数
- 字符串处理函数
- 正则表达式

6.6 习题

6.1 字符串

字符串常数

字符串与字符串常数

- **字符串**(String)是由数字、字母或者其它符号组成的一串字符。
- SPL中，**字符串常数**是指类型为字符串的常数。
- SPL中，常数可以直接写入单元格，此时数据类型是**自动解析**的。如果数据无法识别为数值、布尔值、时间日期等类型，就会被解析为字符串。

	A	B
1	Name	3rd
2	user@website.com	24*3

注意

- 上面例子中，各个单元格中的常数均会被解析为字符串。B1中**既包含数字又包含字母**；而B2中的字符串虽然形式像一个表达式，但**没有用等号作为开头**，所以都会被认为是字符串常数。

字符串常数

字符串常数格

- 有一些情况下，字符串有可能会被自动解析为其它类型，这时可以使用**字符串常数格**来定义字符串常数。
- SPL中，**字符串常数格**是以**半角单引号 ' **开头的，这个单引号只表示这个单元格中常数的类型是**字符串**，不会出现在格值中。
- 字符串常数格的规则**仅使用在定义常数格时。**

	A	B
1	'13100000000	'true
2	'null	

各个单元格中字符串常数的值如下：

Value
13100000000

Value
true

Value
null

注意

- 上面例子中，由于格串都是以 ' 开头的，所以各个格子都是**字符串常数格**。A1不会被识别为长整数，B1不会被识别为布尔值，A2也不会被识别为空值。
- 字符串类型的值在显示时会用**下划线**标识，同时**靠左对齐**。

字符串操作符

- 除了使用字符串常数，在**表达式中**还可以用字符串操作符来定义一个字符串。
- SPL表达式中，常用的**字符串操作符**是一对半角双引号"..."，引号内的字符构成字符串。
- 还有一种**字符串操作符**是**$[...]**，可以和双引号操作符类似使用。

	A	B
1	="3.1416"	=$[3.1416]
2	="doctor"	="I am a "/A2

A1,B1,A2和B2计算后的结果如下：

Value
3.1416

Value
3.1416

Value
doctor

Value
I am a doctor

注意

- 双引号内部的字串在表达式中会作为字符串类型的数据参与计算。B1中的$[...]操作符可以和双引号类似使用。
- B2中使用除号/作为**字符串连接符**，此时要求操作数中**至少有一个是字符串**；字符串连接符也可以用加号+，此时要求**两个操作数全部是字符串**。

特殊的字符串

- 有时候，字符串可能以等号、大于号等字符开头，也可能包含双引号等特殊字符或者关键字，**无法自动解析**，或者**无法正常使用字符串操作符**，需要一些特别的处理。
- 使用**字符串常数格**，可以避免特殊字符串的解析异常的问题，但只能用来定义常数。
- 如果字符串中包含双引号等字符时，直接在**表达式**中用字符串操作符表示会造成解析错误，可以使用**字符串常数格**来避免。

	A	B	C
1	'"I live here."	'=== dividing line ===	'>100
2	="He said, "+C1	="He said, "+""I live here.""	

A1,B1和C1三个字符串常数格的值如下：

Value
"I live here."

Value
=== dividing line ===

Value
>100

A2将两个字符串连接后结果如下：

Value
He said, "I live here."

注意

- 例子中B2在尝试计算时会报错，这是想使用的字符串："I live here."中含有双引号，造成了表达式解析异常。

转义符与特殊字符

- 在**表达式**中直接用字符串操作符定义字符串时，如果字符串中包含"或\字符，可以在它之前用**转义符 ** 标记，用\"或\\表示。
- 如果字符串中包含多个"或\字符，在每一个这样的字符前，都需要添加一个转义符。
- 转义符与一些字母组合时，会构成一些特殊字符，包括**换行符\n**和**制表符\t**，换行符可以使字符串折行显示，制表符可使其后面的文字在一些固定的位置对齐。

	A
1	="He said, "+"\"I live here.\""
2	="C:\\Program Files\\"+"Common Files"

表达式计算后，A1和A2中值如下：

Value
He said, "I live here."

Value
C:\Program Files\Common Files

注意

- 通过在表达式的字符串操作符内添加转义符，能够使得特殊字符串正常解析，避免错误。

换行符的使用

	A
1	="Tom\nJerry"
2	>output(A1)

A1中的\n表示换行符，A2中输出后的结果如下：

注意

- 换行符并不能在右侧的值显示区中表示，只能在输出字符时生效。

制表符的使用

	A
1	="1\t22\t333\t4444\n55555\t666666\t\t88888888"
2	>output(A1)

A1的字符串中除了制表符还包含换行符\n，会输出两行文本，A2输出如下：

1	22	333	4444
55555	666666		88888888

可以看到制表符后字符的对齐情况，第2行中连续两个制表符的效果。

注意

- 制表符同样并不能在右侧的值显示区中表示，**只能在输出字符时生效**。
- 跟在制表符后的字符会在一些**固定的位置**显示，这些位置在不同的软件中显示也不同，可以用多个制表符让其后的字符右移多个固定位置。
- 制表符通常在**多行文本**中使用，以便对齐。

6.2 字符串的转换

字符与ASCII码

- ASCII(American Standard Code for Information Interchange)即**美国信息交换标准代码**，是基于英文字母的一套电脑编码系统（见下页）。
- **基础ASCII码**将显示英语字符时所需的字母、数字、符号以及一些特殊字符转换为**0~127**的代码。
- 除了基础ASCII码，还有很多扩充的**编码标准**，用来支持各种语言的显示。

	A	B
1	=asc("a")	=asc("Z")
2	=char(43)	=char(B1)

A1,B1,A2和B2计算后的结果如下：

Value
97

Value
90

Value
+

Value
Z

注意

- 获得字符**c**的ASCII码用**asc(c)**函数，将ASCII码**n**转换为对应的字符用**char(n)**函数。
- 程序中的**函数**是指一些有着特定名称的方法，能够根据输入的参数执行处理或返回结果。有关函数的概念将在**第7章**中具体说明，本章只需了解字符串相关的一些函数如何使用就可以了。

ASCII码表

码值	字符	码值	字符	码值	字符	码值	字符	码值	字符	码值	字符	码值	字符	码值	字符
0	NUL	16	DLE	32	(space)	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	|
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

【例6.1】随机生成一个4个英文字符构成的字符串，首字母大写，其它小写。

【例6.1】随机生成一个4个英文字符构成的字符串，首字母大写，其它小写。

解题思路

用ASCII码转换为所需字符，使用随机函数**rand(n)**能够随机生成0~n-1的整数。

	A	B	C
1	4	=char(65+rand(26))	/先在B1中随机生成首字母
2	for A1-1	>B1+=char(97+rand(26))	/循环，添加后面的随机字母

每次执行都会在B1中随机生成一个字符串，如：

Value
Rfau

注意

- 从前面的基本ASCII码表中可以知道，**大写英文字母**的编码依次对应**65~90**，**小写英文字母**依次对应**97~122**。

编码与转换

- 基础ASCII码只能用来显示英文，如果需要显示其它语言或者其它字符，就需要使用扩充的编码标准，也称为**字符集**。
- 字符集的体系很多，如中国国家标准**GB**，国际标准**ISO**等。每个体系内又有不同的字符集，如汉字常用的字符集有**Unicode**，**GBK**和**GB2312**等。
- 在将汉字等非英文字符与对应的编码相互转换时，**必须明确使用的字符集**。
- Unicode又称统一码，是能够满足**跨语言、跨平台**文本转换的字符集，最为通用。

	A	B
1	=asc("字")	=char(A1)
2	=urlencode("字符串", "Unicode")	=urlencode@r(A2, "Unicode")
3	=urlencode("字符串", "GBK")	=urlencode@r(A2, "GBK")

各个单元格计算的结果如下：

Value
23383

Value
字

Value
%FE%FF%5B%57%7B%26%4E%32

Value
字符串

Value
%D7%D6%B7%FB%B4%AE

Value
字符串

注意

- 前面介绍的**asc(c)和char(n)**函数，实际上是**将字符和Unicode编码相互转换**，但Unicode是兼容基础ASCII的，因此c既可以是英文字符，也可以是汉字等其它字符。
- 为了查看字符s的不同编码转换，可以借用url串的转码函数**urlencode(s, cs)**，其中cs是使用的字符集，转码后的编码c可以用**urlencode@r(c, cs)**逆转换为字符。

常见字符集与语言

字符集名称	语言
ASCII	英语，西欧语
ISO-8859-1	英语，西欧语
GBK	中文简体，中文繁体
GB2312	中文简体
BIG5	中文繁体
GB18030	中文，日文，朝鲜语
UTF-8	各国语言
Unicode	各国语言

显示格式

- 无论任何类型的数据，如整数、布尔值、时间日期等，在显示时都会显示成一个字符串。如果需要让数据按照所需的效果显示，可以设置数据的**显示格式**。
- **显示格式**是一个字符串，可以按照它的规则将某种类型的数值转换为字符串，如浮点数的一个显示格式**"$#,##0.00"**。在这个显示格式串中，有的是占位符，如**"#"**和**"0"**，表示数字的显示方式；有的是普通字符，如**"$"**，表示输出一个字符；有的是功能符，如小数点**"."**和分节号**","**，表示数据有小数点或需要分节。

	A	B
1	'$#,##0.00	=string(1234567.8, A1)
2	'MM/dd/yyyy	=string(date(2020,1,20), A2)

B1,B2中数据按显示格式转换后如下：

Value
$1,234,567.80

Value
01/20/2020

注意

- A1,A2中用字符串常数格存储了两个显示格式，A1中的显示格式通常用来显示金额，B1中的显示格式通常用来显示日期，转换显示格式使用**string**函数。
- B2中的**date(y,m,d)**函数按照年月日参数生成一个日期数据。
- 转换后，B1中得到的字符串以$开头，金额的整数部分每3位一节用逗号**分节**，小数部分显示两位，位数不够用0补足。B2中的数据按**月/日/年**的格式显示。

数值型数据显示格式中的常用字符

显示格式中字符	作用
0	显示一位数字，如果缺位则补0
#	显示一位数字，如果缺位则该位不显示
.	小数点，分隔实数的整数和小数部分。当显示结果无小数部分时，小数点不显示
,	分节符，在显示整数部分时使用，表示每若干位用逗号分隔，通常为每3位一节
E	科学计数法表示，在E的前面表示尾数的格式，后方表示指数的格式
%	百分号，显示百分比数据，将数值乘100后按百分比显示
‰	千分号，显示千分比数据，将数据乘1000后按千分比显示
;	显示子模式分隔符，分隔两个显示格式，分隔符后表示负数所用的符号
-	负号，显示负号。一般不需要使用
'…'	用于将某些可能产生歧义的字符括起来表示需要显示

日期型数据显示格式中的常用字符

显示格式中字符	作用
y/yy	年，两位数字
yyyy	年，四位数字
M	月份
MM	月，两位数字，不足用0补齐
MMM	月，英文环境下为英文简写
MMMM	月，英文环境下为英文全称
d	日
dd	日，两位数字，不足用0补齐
E	星期，英文环境下为英文简写
EEEE	星期，英文环境下为英文全拼
G	Era标识符，公元前/公元后，缩写
w	本年的第几周
W	本月的第几周
D	本年的第几天

时间型数据显示格式中的常用字符

显示格式中字符	作用
H	小时，24小时制，0~23
HH	小时，24小时制，00~23，两位数字，不足用0补齐
k	小时，24小时制，1~24
kk	小时，24小时制，01~24，两位数字，不足用0补齐
h	小时，12小时制，1~12
hh	小时，12小时制，01~12，两位数字，不足用0补齐
K	小时，12小时制，0~11
KK	小时，12小时制，00~11，两位数字，不足用0补齐
m	分钟
mm	分钟，两位数字，不足用0补齐
s	秒
ss	秒，两位数字，不足用0补齐
S	毫秒
a	上下午AM/PM
z	时区

6.2 字符串的转换

【例6.2】2020年1月1日是周三，请输出前三个月的月历。在月历中，每周占一行，从周日开始显示。

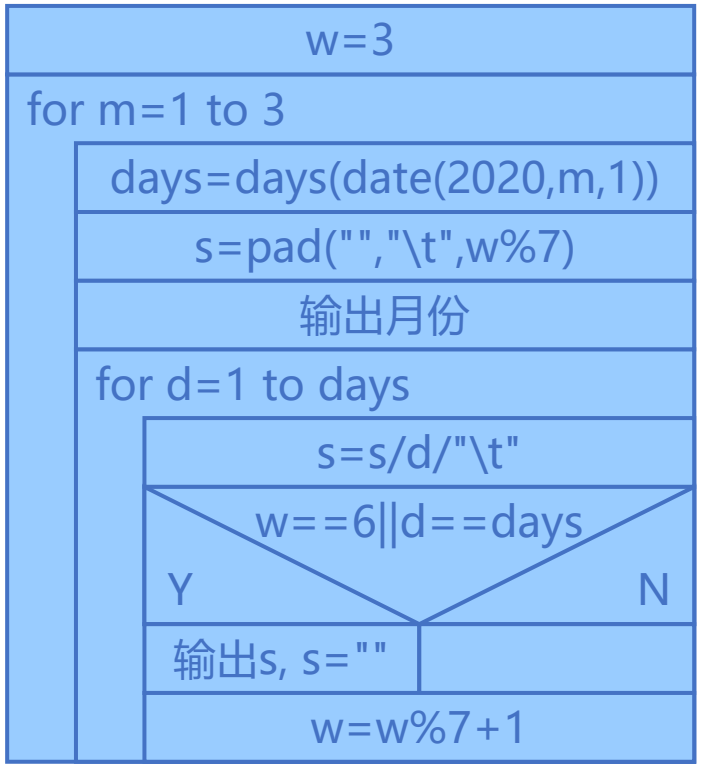
解题思路

可以用days函数计算某个月的天数，如计算2020年3月的天数，可以用days(date(2020,3,1))。

显示每月的月历时，可以先用显示格式输出月份。

每周的日期中间用制表符\t分隔。

第一周需要先根据1日的星期填入多个\t，此时可以用pad函数，pad(s,c,l)函数可以在串s前补充多个字符c，使得s的总长度达到l。



6.2 字符串的转换

【例6.2】2020年1月1日是周三，请输出前三个月的月历。在月历中，每周占一行，从周日开始显示。

	A	B	C	D	E
1	3				/初始化星期
2	for 3	=date(2020,A2,1)	=days(B2)	=pad("","\t",A1%7)	/本月天数，首周补足空白天数
3		=string(B2,"MMMM")	>output(B3)		/输出月份
4		for C2	>D2=D2/B4/"\t"		/添加本日日期
5			if A1==6 || B4==C2	>output(D2)	/周末或月底，输出
6				>D2=""	/输出后清空字串
7			>A1=A1%7+1		/星期加一

结果：

16	17	18	19	20	21	22
23	24	25	26	27	28	29
March						
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

按显示格式解析数据

- 如果初始数据是按照**显示格式**存储的，也可以指定显示格式来解析。
- 按格式解析数据可以用**number(s,fmt)，date(s,fmt)，time(s,fmt)**和**datetime(s,fmt)**这几个函数，函数中**s**为数据字符串，**fmt**为对应的显示格式串。
- 解析数据时需要根据数据的类型是数值型、日期型、时间型或日期时间型，选取不同的解析函数。

	A	B
1	'$#,##0.00	=number("$1,234,567.80", A1)
2	'MM/dd/yyyy	=date("01/20/2020", A2)

B1,B2中得到的数据如下：

Value
1234567.8

Value
2020-01-20

注意

- "$1,234,567.80 "并不是标准格式的数值，因此无法直接识别，需要用**number**函数指明显示格式才能解析为浮点数。
- 日期时间的格式非常多，在不符合默认格式的情况下，同样需要指明显示格式才能够用对应的解析函数识别。

6.3 动态程序

宏

- SPL中的宏，是指用来**组成表达式**的**字符串**，这个字符串**通常是动态生成的**。
- 宏用**${...}**表示，括号中是一个表达式或变量，计算结果是一个字符串。
- 在字符串操作符"…"的引号内部，宏是无效的。

	A	B
1	+	-
2	=3${A1}4	=3${B1}4

A2和B2计算后结果如下：

Value
7

Value
-1

注意

- 上面例子中，A2和B2中使用了宏，用单元格变量中的符号拼成表达式计算，A2中相当于计算3+4，B2中相当于计算3-4。

eval函数

- SPL中的eval(x)函数，可以将参数**x**计算得到的**字符串**作为表达式**重新解析**，并计算出结果。
- 宏只组成表达式的一部分，而eval函数会解析一个新的表达式并计算。
- **eval(x, ...)**函数的表达式串中可以使用参数，参数在x后输入，表达式串中的参数用?表示，多个参数用**?n**表示。

	A	B
1	10/2	(?1+?2)/2
2	=eval(A1)	=eval(B1, 3, 7.8)

A2和B2计算后结果如下：

Value
5.0

Value
5.4

注意

- A2和B2中使用了eval函数计算上面的两个表达式。其中A2计算一个除法算式，而B2计算两个数的平均值，这两个数在eval函数中依次输入，在原表达式中用?1和?2表示。

eval函数中用$[...]表示字符串

在6.1字符串讲到字符串操作符时，提到SPL中有两种字符串操作符，"..."和$[...]，在eval函数中或者宏中使用时，两种字符串表示法有一些区别：

	A	B
1	10	23
2	25	75
3	=eval("A1+B1+A2+B2")	=eval($[A1+B1+A2+B2]

	A	B	C
1	10		23
2			
3	25		75
4	=eval("A1+B1+A2+B2")		=eval($[A1+C1+A3+C3])

上面的网格代码中，A3和B3根据一个表达式字符串，计算A1,B1,A2,B2四个单元格变量的和，它们执行的效果是相同的。

但是，如果需要在两列代码之间插入一列，在前两行代码之间插入一行，网格会变成右边的网格。此时C4中的表达式仍然正确，而A4中的表达式就出了无法对应的问题。

注意

- 从上面例子可以看到，虽然两种字符串操作符都能够表示一个表达式字符串，但是$[...]的内容中出现的单元格名，将会随着网格的编辑操作自动调整，而双引号内的字符串不会变化。

【例6.3】随机生成5道口算题，要求题目是两个两位数之间的加、减或乘法，答案在题目全部输出后再给出。

eval函数

【例6.3】随机生成5道口算题，要求题目是两个两位数之间的加、减或乘法，答案在题目全部输出后再给出。

解题思路

用**rand(90)+10**能够随机生成1个两位数，符号可以用宏随机从单元格常数中获取，答案需要用eval函数计算。

	A	B	C	D
1	+	-	*	/运算符号
2	=""			/初始化答案
3	for 5	=rand(90)+10	=rand(90)+10	/循环，生成随机两位数
4		=char(65+rand(3))	=B3/${B4}1/C3	/随机生成一道题目
5		>A2=A2/eval(C4)/","	>output(C4)	/记录答案，输出题目
6	>output(A2)			/输出答案A2

输出结果如下：

```
12+52
86+44
93+59
34+63
34+85
64,130,152,97,119,
```

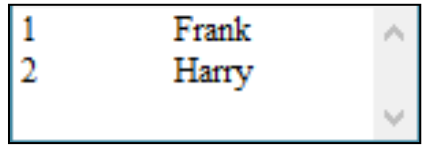
▶ B4中先随机生成一个A~C的字母，C4中再利用宏来获取一个符号，生成题目。如B4中得到"A"，那么C4中相当于计算=B3/A1/C3，将两个两位数和中间的符号用字符连接符连起来。

6.4 文本文件

6.4 文本文件

文本文件

- **文本文件**是以**文本方式**存储的计算机文件，文件中存储的数据由**若干行字符串**构成。
- 文本文件通常使用"**txt**"作为扩展名。还有一些其它扩展名的文本文件，如在第10章中将提到的csv文件等。
- 一个文本文件存储时需要采用一致的编码方式，常用的字符集是**Unicode**，也可能采用其它字符集存储。如果打开文本文件时选择了错误的编码方式，有可能出现**乱码**，无法读出正确的字符。



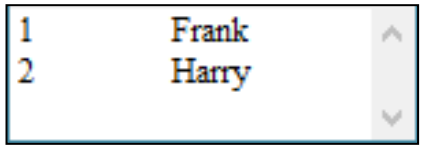
6.4 文本文件

文本文件的输入或输出

- SPL中，从文本文件中读取数据使用f.read()函数。
- SPL中，可以用f.write(s)函数将一个字符串写入文本文件；也可以添加选项，用f.write@a(s)函数将字符串s写到文本文件已有内容的后面。

	A	B
1	D:\files\info.txt	
2	>file(A1).write("1\tFrank")	>file(A1).write@a("2\tHenry")
3	=file(A1).read()	

A2和B2向文本文件中写入两行数据：



注意

- txt文件是使用非常广泛的一类文本文件，在Windows等系统中，可以双击它用**记事本**之类软件查看文件的内容。
- f.read()函数将读入文本文件中的所有字符，而其中的换行符、制表符等特殊字符是无法在右侧的值显示区查看的。

【例6.4】从文本文件paragraph.txt中，读取英文文本，统计其中字母出现e的次数。

At heart he was an arrant coward, which is the way with bullies among apes as well as among men; so he did not remain to fight and die, but tore himself away from them as quickly as he could and fled into the sheltering boughs of the forest.

6.4 文本文件

【例6.4】从文本文件paragraph.txt中，读取英文文本，统计其中字母出现e的次数。

At heart he was an arrant coward, which is the way with bullies among apes as well as among men; so he did not remain to fight and die, but tore himself away from them as quickly as he could and fled into the sheltering boughs of the forest.

解题思路

根据文件名，用**file(fn).read()**读取文件中的字符串，再循环从字符串中选取字符，判断字符是否为"e"。循环时，计算字符串长度使用**len(s)**函数，选取字符时，使用**mid(s, n, 1)**函数。

	A	B	C	D
1	=file("paragraph.txt").read()	0		/从文件中读取字符串
2	for len(A1)	=mid(A1, A2, 1)		/循环从字符串中取字符
3		if B2=="e"	>B1=B1+1	/如果字符是e，计数
4	>output(B1)			/输出结果

结果如下：

【例6.5】请计算下面3道四则运算题目，并将结果输出到文件 D:\files\answers.txt。

- 25*8*4*1.25
- (107-1)/2+9
- 123456789*8+9

6.4 文本文件

【例6.5】请计算下面3道四则运算题目，并将结果输出到文件 D:\files\answers.txt。

- 25*8*4*1.25
- (107-1)/2+9
- 123456789*8+9

解题思路

用eval()函数可以计算表达式的结果，用file.write()函数将文本写到文件。

	A	B	C	D
1	25*8*4*1.25	(107-1)/2+9	123456789*8+9	/输入3个表达式
2	D:\files\answers.txt			/输出文件路径
3	>A1+="="/eval(A1)	>B1+="="/eval(B1)	>C1+="="/eval(C1)	/计算各个表达式，添加结果
4	>file(A2).write(A1)	>file(A2).write@a(B1)	>file(A2).write@a(C1)	/输出结果

输出文件中结果如下：

```
25*8*4*1.25=1000.0
(107-1)/2+9=62.0
123456789*8+9=987654321
```

6.5 字符串函数

字符函数

- SPL中的字符函数，是指用来执行**字符串转换，字符串类型判断**等功能的函数。
- 本章前面使用到的asc, char, urlencode, string等函数都是用来执行字符串转换的字符函数。

	A	B
1	'$#,##0.00	=string(1234567.8, A1)
2	=asc("Z")	=isalpha("Alphabet")

注意

- 上面例子中，B1根据显示格式将数值转换为字符串，A2计算字符的编码，B2判断字符串是否全部由字母构成。

常用字符函数

函数	作用
asc(s)	字符串s第1个字符的Unicode编码
asc(s, n))	字符串s第n个字符的Unicode编码
char(n))	编码为n的字符
isdigit(s)	字符串s是否全部由数字组成
isalpha(s)	字符串s是否全部由字母组成
islower(s)	字符串s是否全部由小写字母组成
isupper(s)	字符串s是否全部由大写字母组成
lower(s)	把字符串s中的字母变为小写
upper(s)	把字符串s中的字母变为大写
string(x, fmt)	按照显示格式fmt转换数据
urlencode(s, cs)	按照字符集cs转换url字串
urlencode@r(s, cs)	按照字符集cs逆转换url字串

字符串处理函数

- SPL中的**字符处理函数**，是指用来**处理字符串**，或者执行**字符串相关计算**等功能的函数。
- 字符串处理函数可以计算字符串长度，修改字符串，截取字符串，生成字符串等等。

	A	B
1	Helen Keller	=len(A1)
2	=left(A1, 5)	=replace(A1, "e", "E")

注意

上面例子中，B1计算字符串的字符数，A2截取字符串中的左边5个字符，B2将字符串中的小写字母e全部换为大写。

常用字符串处理函数

函数	作用
len(s)	字符串s的长度，即s的字符总数
fill(s, n))	n个字符串s连接成一个长字符串
left(s, n))	字符串s左侧的n个字符构成的字符串
right(s, n)	字符串s右侧的n个字符构成的字符串
mid(s, n, len)	字符串s中从第n个起的len个字符构成的字符串
pos(s_1, s_2)	在字符串s_1中查找子串s_2所在位置，找不到返回null
replace(s, a, b)	字符串s是否全部由大写字母组成
like(s, fmt)	字符串s是否满足fmt描述的格式
trim(s)	去掉字符串s两侧的空格
pad(s, c, len)	在字符串s前添加字符c，直至字符长度到len
rands(s, len)	用字符串s中的字符，随机生成一个长为len的串

【例6.6】输入一个英文全名，输出右侧的姓氏部分，如输入Helen Keller。

【例6.6】输入一个英文全名，输出右侧的姓氏部分，如输入Helen Keller。

解题思路

要取出最右侧的一个单词，需要先查找最右侧空格所在的位置，再取出这个位置右侧的所有字符构成的字符串，从右侧查找字符位置，使用**pos@z()** 函数。

	A	B
1	Helen Keller	/输入全名
2	=pos@z(A1, " ")	/最后一个空格的位置
3	>output(right(A1,A2))	/输出姓氏部分

输出的姓氏部分如下：

Keller

正则表达式

- **正则表达式**(Regular Expression)，又称规则表达式，描述文本的某个构成规则的字符串。
- SPL中，用**s.regex(re)**函数，可以用正则表达式**re**来匹配字符串**s**，如**s**能够匹配正则表达式，则取出**re**中用小括号标记的部分。
- SPL中使用regex函数时，正则表达式中可以有**多个**括号标记的取出部分，各个取出的字符串将构成**序列**，有关序列的概念，将在第8章中详细介绍。

	A	B
1	a1253b	
2	=A1.regex("(a[0-9])")	=A1.regex("a([0-9]*)")

A2和B2计算后结果如下：

Index	Member
1	a1

Index	Member
1	1253

注意

- 上面例子中，A2中正则表达式“(a[0-9])”表示以字母a开头后面接1位数字，并取出这部分字符串；B2中的正则表达式“a([0-9]*)”表示字符串以字母a开头并取出它后面连续的数字。

正则表达式中常用的符号

符号	意义
.	小数点表示任何一个字符
(…)	获取匹配括号内部分的字符串
x|y	匹配x或者y
[a-z]	a至z范围内的任一个字符，a和z可以改变，如[0-9]表示0~9之间的任一个数字
[^a-z]	a至z范围之外的任一个字符
[xyz]	匹配x,y,z中任一个字符
[^xyz]	匹配任何一个x,y,z以外的字符
*	前面的子表达式连续任意多个
?	前面的子表达式1个或0个
{n}	前面的子表达式连续n个
\d	任一个数字字符，需要加转义符为\\d
\D	任一个非数字字符，需要加转义符为\\D
\t	匹配一个制表符，需要加转义符为\\t
\s	匹配一个不可见字符，如空格，制表符，换行符等，需要加转义符为\\s
\S	匹配一个可见字符，需要加转义符为\\S
\uXXXX	匹配一个Unicode字符，XXXX为四个十六进制数字，需要加转义符为\\uXXXX

正则表达式的规则难度较大，这里只作初步了解，应用时还需要深入查阅相关资料

6.6 习题

6.6 习题

【习题6.1】将输入的3个单词全部转为大写字母，输出到文件。如输入Quick,brown和foX。

【习题6.2】在june.txt中输出2020年6月的月历。在月历中，每周占一行，从周日开始显示，第一行显示英文月份，第二行显示星期缩写。2020年6月1日为周一。

【习题6.3】输入一段英文，将它们转换为ASCII码输出到code.txt中，每个字符对应一个3位的编码。如输入：We know only that on a bright May morning in 1888, John, Lord Greystoke, and Lady Alice sailed from Dover on their way to Africa.

【习题6.4】从上一道题的code.txt文件中读出编码，并将它恢复为英文。

【习题6.5】输入一个英文名，用正则表达式从中取出名字部分（First Name）。如输入Helen Keller.

【习题6.6】输入一句英文，依次输出句子中的每个英文单词。如输入： The quick brown fox jumps over a lazy dog.

【习题6.7】随机生成1道计算题，要求题目是3个三位数之间的加、减运算，答案用脱式格式给出，输出到题目中，后面一步有可能出现括号。即格式如：

XXX+(XXX-XXX)

=XXX+XXX

=XXX

6.6 习题

【习题6.1】将输入的3个单词全部转为大写字母，输出到文件。如输入Quick,brown和foX。

6.6 习题

【习题6.1】将输入的3个单词全部转为大写字母，输出到文件。如输入Quick,brown和foX。

	A	B	C	D
1	Quick	brown	foX	/输入3个单词
2	D:\files\words.txt			/输出文件路径
3	>file(A2).write(upper(A1))	>file(A2).write@a(upper(B1))	>file(A2).write@a(upper(C1))	/输出结果到文件

结果：

```
QUICK
BROWN
FOX
```

6.6 习题

【习题6.2】在june.txt中输出2020年6月的月历。在月历中，每周占一行，从周日开始显示，第一行显示英文月份，第二行显示星期缩写。2020年6月1日为周一。

6.6 习题

【习题6.2】在june.txt中输出2020年6月的月历。在月历中，每周占一行，从周日开始显示，第一行显示英文月份，第二行显示星期缩写。2020年6月1日为周一。

	A	B	C	D
1	1	=date(2020,6,1)	=days(B1)	/本月天数
2	=""	=file("D:/files/june.txt")	>B2.write(string(B1, "MMMM"))	/输出月份
3	for 7	=pdate(B1)+A3-1	>A2=A2/string(B3, "E")/"\t"	/生成星期行数据
4	>B2.write@a(A2)	>A2=pad("","\t",A1%7)		/输出星期行，首周补足空白天数
5	for C1	>A2=A2/A5/"\t"	>A1=A1%7+1	/添加本日日期，设置明日星期
6		if A1==7 || A5==C1	>B2.write@a(A2)	/周末或月底，输出本周行
7			>A2=""	/输出后清空行字串

结果：

June

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

注意

- pdate(date)函数，可以用来获取date所在周的第1天，国际惯例是把周日作为每周的第一天的。

6.6 习题

【习题6.3】输入一段英文，将它们转换为ASCII码输出到code.txt中，每个字符对应一个3位的编码。如输入：We know only that on a bright May morning in 1888, John, Lord Greystoke, and Lady Alice sailed from Dover on their way to Africa.

6.6 习题

【习题6.3】输入一段英文，将它们转换为ASCII码输出到code.txt中，每个字符对应一个3位的编码。如输入：We know only that on a bright May morning in 1888, John, Lord Greystoke, and Lady Alice sailed from Dover on their way to Africa.

	A	B	C
1	We know only that on a bright May morning in 1888, John, Lord Greystoke, and Lady Alice sailed from Dover on their way to Africa.	=""	/输入英文段落，初始化输出文本
2	for len(A1)	=mid(A1, A2, 1)	/按字符总数循环，每次取出一个字符
3		=asc(B2)	/将字符转换为ASCII编码
4		>B1=B1+string(B3, "000")	/添加一个字符的编码，显示格式000表示3位整数不足补0
5	>file("D:/files/code.txt").write(B1)		/输出ASCII编码到文件

结果：

```
087101032107110111190321111101081210321161040971160321 1
111003209703209811410510310411603207709712103210911111411
010511010303210511003204905605605604403207411110411004 4
032076111114100032071114101121115116111071010440320971 1
010003207609710012103206510810509910103211509710510810 1
100032102114111109032068111118101114032111100321161041 01
105114032119097121032116111032065102114105099097046
```

【习题6.4】从上一道题的code.txt文件中读出编码，并将它恢复为英文。

6.6 习题

【习题6.4】从上一道题的code.txt文件中读出编码，并将它恢复为英文。

	A	B	C
1	=file("D:/files/code.txt").read()	=""	/从文件中读取文本，初始化解码文本
2	for len(A1)/3	=mid(A1, A2*3-2, 3)	/循环次数按编码总字符的三分之一计算，每次取出3个字符，是一个完整编码
3		=number(B2)	/将读出字符转换为编码值
4		>B1=B1+char(B3)	/解码一个字符，记录
5	>output(B1)		/输出解码结果

结果：

We know only that on a bright May morning in 1888, John, Lord Greystoke, and Lady Alice sailed from Dover on their way to Africa.

6.6 习题

【习题6.5】输入一个英文名，用正则表达式从中取出名字部分（First Name）。

如输入Helen Keller.

6.6 习题

【习题6.5】输入一个英文名，用正则表达式从中取出名字部分（First Name）。如输入Helen Keller.

	A	B
1	Helen Keller	/输入英文名
2	=A1.regex("(\\S*)\\s")	/用正则表达式取出First Name，即首个非可见字符\s前的部分

结果：

Index	Member
1	Helen

6.6 习题

【习题6.6】输入一句英文，依次输出句子中的每个英文单词。如输入： The quick brown fox jumps over a lazy dog.

6.6 习题

【习题6.6】输入一句英文，依次输出句子中的每个英文单词。如输入： The quick brown fox jumps over a lazy dog.

	A	B	C	D
1	The quick brown fox jumps over a lazy dog.	=""		/输入英文，初始化单词记录
2	for len(A1)	=mid(A1, A2, 1)		/循环取出每个字符
3		if isalpha(B2)	>B1=B1+B2	/如果字符是英文字母，记录到单词
4		else	>output(B1)	/字符非英文字母，输出单词
5			>B1=""	/单词记录清空

结果：

```
The
quick
brown
fox
jumps
over
a
lazy
dog
```

6.6 习题

【习题6.7】随机生成1道计算题，要求题目是3个三位数之间的加、减运算，答案用脱式格式给出，输出到题目中，后面一步有可能出现括号。

6.6 习题

【习题6.7】随机生成1道计算题，要求题目是3个三位数之间的加、减运算，答案用脱式格式给出，输出到题目中，后面一步有可能出现括号。

	A	B	C	D
1	+	-	=char(65+rand(2))	/记录符号，C1随机选符号的列(A或B)
2	=rand(900)+100	=rand(900)+100	=rand(900)+100	/随机生成3个三位数
3	=A2/${C1}1/B2	=eval(A3)	>C1=char(65+rand(2))	/用宏生成第一步计算的式子，C1随机选第二步的符号所在列
4	if rand(2)==0	>A3=A3/${C1}1/C2	>B3=B3/${C1}1/C2	/不需括号，第二步顺序计算，生成完整式子和第二步结果
5	else	>A3=C2/${C1}1/"("/A3/")"		/第一步实际在后方，生成完整式子时需加括号
6		if B3<0	>B3=C2/${C1}1/"("/B3/")"	/第一步结果在后方时，需要考虑第一步计算后是不是负数，负数时要加括号
7		else	>B3=C2/${C1}1/B3	/第一步计算后不是负数，正常生成第二步结果
8	>output(" "/A3)	>output("="/B3)	>output("="/eval(B3))	/输出每一行脱式结果

结果：

```
960+(471-479)
=960+(-8)
=952
```

第7章 函数与子程序

7.1 什么是函数

7.2 在SPL中调用函数

- 函数的调用
- SPL中的函数参数
- SPL中的函数选项

7.3 子程序

- 子程序的定义及调用
- 子程序中参数的传递
- 变量的作用范围
- 子程序的调用过程
- 子程序的嵌套
- 子程序的递归

7.4 跨网格调用

- 跨网格调用
- 跨网格调用参数的传递
- 跨网格调用的嵌套和递归

7.5 习题

7.1 什么是函数

7.1 什么是函数

什么是函数

- 计算机程序中的**函数**就是功能，每个函数用来实现一个特定的功能。
- 通过**函数名**可以执行函数，函数的名称往往代表着它的功能。
- 在程序中执行函数，称为**调用函数**。调用函数时，需要输入的数据称为**参数**。
- 函数根据是否需要输入数据，分为**无参函数**和**有参函数**。

	A	B
1	3.1416	3
2	=if(A1<B1, B1, A1)	

注意

- A2中用到的if(a,b,c)函数在第4章中已经介绍过了，这就是一个函数，函数名是**if**。它的功能是根据a是否成立，计算b或c作为结果返回。执行后，A2中得到的值为A1，B1中的较大值。

为什么要用函数

- 函数能够实现特定的功能，通过调用函数，可以避免重复编写程序，**有效降低工作量**。
- 高级语言都会将一些常用的功能模块编写为函数，用户可以直接使用，既可以**提高效率**，又增加了程序的**可靠性**。

7.2 在SPL中调用函数

函数的调用

- 函数的调用很简单，只需在表达式中通过**函数名**调用即可。
- 函数一般在调用时需要输入**参数**，也有一些函数可以不输入参数，如**pi()**和**now()**。
- 有的函数有返回值，也有一些函数并没有返回值，而只是改变某些参数。
- 在SPL中，**同样名称的函数**能够通过增加**函数选项**或**改变输入参数**，提供多种功能模式。

	A	B
1	=pi()	=pi(5)
2	=now()	

注意

- A1中用到的圆周率函数**pi()**未使用参数，此时结果为π；而B1中输入了参数5，此时结果为5π.
- A2中用到的当前时刻函数**now()**，将返回当前时刻的**日期时间**。
- 在一段程序中，同一个函数可以被调用任意多次。

【例7.1】输入圆的半径（如5.5），计算出圆的周长和面积。

7.2 在SPL中调用函数

【例7.1】输入圆的半径（如5.5），计算出圆的周长和面积。

解题思路

使用刚刚介绍过的函数pi(n)，计算出结果。

	A	B	C
1	5.5		/输入半径
2	=2*pi()*A1	=pi(A1*A1)	/用pi函数计算周长和面积

A2计算出圆的周长，B2计算出圆的面积，选择单元格可在右侧查看结果：

Value
34.55751918948772

Value
95.03317777109125

注意

- A2中使用了无参数的pi()函数，相当于直接返回圆周率。A2中的表达式也可以改为=pi(2*A1)，结果是相同的。
- 函数的参数可以直接输入数值，也可以用表达式来计算。

在前面的章节中，我们已经介绍过字符串相关函数，if函数等，这里再来了解一下常用的日期时间函数：

常用日期时间函数

函数	作用
date(y, m, d)	指定年月日生成日期类型数据
time(h, min, s)	指定时分秒生成时间类型数据
datetime(y, m, d, h, min, s)	指定年月日时分秒生成日期时间类型数据
now()	当前时刻，日期时间类型，精确到毫秒
pdate(dt)	与指定日期同段的第1天，默认为本周第1天
days(dt)	指定日期所在月的天数
age(dt)	按照生日计算当前的年龄，默认精确到日
elapse(dt, n)	默认计算指定日期时间n天后的日期时间
interval(dt1, dt2)	默认计算两个日期时间的间隔天数

SPL中函数的参数

- 不同的函数有不同的**参数个数**和**参数类型**要求，调用函数时输入。
- SPL中，如果函数需要输入多个参数，参数之间需要用**逗号,分号;**或者**冒号:**分隔开，要注意函数对分隔符的不同要求。
- 有些函数采用**点语法**，如序列、序表相关的各种函数。

	A	B
1	=round(3.1416)	=round(3.1416, 2)
2	[1,3,5.5,2]	=A2.len()

A1,B1和B2计算后的结果如下：

Value
3.0

Value
3.14

Value
4

注意

- A1和B1中调用的都是四舍五入函数**round()**，不同的是A1中只输入了1个参数，计算它四舍五入到个位的结果，B1中输入了2个参数，参数之间用逗号分隔，第二个参数“2”表示四舍五入的结果保留2位小数。
- 使用**点语法**的函数都是针对某些特定数据类型的功能，例如B2中计算序列成员数的函数**A2.len()**就是针对序列A2的计算。这些函数将在第8章~第11章详细讲解。

SPL中函数参数的分隔

SPL中，有的函数的参数之间可以使用多种分隔符，比如if函数，我们来看下面用来判断体质水平的程序：

	A	B	C
1	77.2	1.78	/输入体重(kg)和身高(m)
2	=A1/B1/B1		/计算bmi指数
3	=if(A2<18.5:"Underweight", A2<24:"Fit", A2<28:"Overweight"; "Obesity")	>output(A3)	/判断体型如何，输出结果

输出结果如下：

```
Overweight
```

注意

- 上面例子中，采用了if函数的标准写法，**if(c_1:v_1, c_2:v_2,...; v_0)**，其中c_1:v_1表示当条件表达式c_1满足时，返回值v_1，这样一组参数之间用冒号分隔；各个条件之间用逗号分隔；而v_0表示前面的条件均不满足时，if函数返回的结果，所以和其它参数用分号隔开。与全使用逗号做分隔符相比，这样的写法读起来更清晰，更准确，而且在书写函数时也更不容易出错。

SPL中的函数选项

- 使用函数选项调用函数的方式是 *f@o(...)*，在函数名 *f* 后添加@符号和选项字符 *o*。
- SPL中，很多函数可以通过添加**函数选项**，提供**不同的功能**。通过使用函数选项，能够在扩充功能模式的同时，避免使用过多的函数名。
- 每一个函数选项表示一种功能模式，多种功能模式有可能需要合并使用，此时可以**同时**使用**多个选项**。

	A
1	=now@t()
2	=now@d()

注意

- A1和A2中用到的函数**now()**都添加了选项，A1中的选项@t，表示仅返回当前的**时间**；A2中的选项**@d**表示仅返回当天的**日期**。

【例7.2】Kevin出生于1992年6月23日，那么，到2020年3月1日为止，他多少岁？此时距他出生多少天，多少月，多少周?

【例7.2】Kevin出生于1992年6月23日，那么，到2020年3月1日为止，他多少岁？此时距他出生多少天，多少月，多少周?

解题思路

计算日期差值，应该使用interval函数，计算差值时需要计算不同的日期单位时，添加不同选项。@y表示以年为单位，@m表示以月为单位，@w表示以周为单位。

	A	B	C
1	=date(1992,6,23)	=date(2020,3,1)	/生成两个日期
2	>output(interval@y(A1,B1)/" years")	>output(interval(A1,B1)/" days")	/输出年差值和日差值
3	>output(interval@m(A1,B1)/" months")	>output(interval@w(A1,B1)/" weeks")	/输出月差值和周差值

输出结果如下：

```
28 years
10113 days
333 months
1444 weeks
```

注意

- 计算日期差值时，当时间进入下个时间段时，计数就会加1，如计算年龄时是以1月1日为界限的，到下一年1月1日就会认为年龄加1。所以，即使2020年生日未到，也会计算为28岁。

7.3 子程序

什么是子程序

- **子程序**是能够实现某种功能的一段程序，子程序能够由主体程序调用，在调用完成后会返回主体程序中继续执行。
- SPL中，子程序是以语句**func**为**主格**的**代码块**，结果需要用**return**语句返回。如果没有return语句，子程序将在代码块执行完毕后返回，此时**无返回结果**。
- SPL中，调用子程序需要使用**func(C,...)**函数，其中**C**是子程序的主格，后面根据需要可以输入子程序所使用的参数。

	A	B	C	D
1	=func(A2)	=func(A2)	=A1*10+B1	/随机生成一个由1~6构成的两位数
2	func	=rand(6)+1		/子程序
3		return B2		/返回1~6的数字

注意

- B2中用到的随机函数**rand(n)**会生成一个0~n-1的随机数，加1后生成一个1~6的随机数。
- A2中用func语句定义子程序，子程序不需输入参数，随机返回一个1~6的数，由B3返回。
- A1和B1中分别调用A2中的子程序生成一个1~6的数，C1中得到由1~6数字随机构成的两位数。

【例7.3】用子程序随机生成10道口算题，要求题目是两个两位数之间的加、减或乘法，答案在每道题目后用等号给出。 输出题目到question.txt文件中，每道题占一行。

【例7.3】用子程序随机生成10道口算题，要求题目是两个两位数之间的加、减或乘法，答案在每道题目后用等号给出。 输出题目到question.txt文件中，每道题占一行。

	A	B	C	D
1	+	-	*	/运算符号
2	=file("question.txt")	>A2.write("")		/初始化输出文件
3	for 10	>func(A4)		/循环10次调用子程序
4	func	=rand(90)+10	=rand(90)+10	/子程序，生成随机两位数
5		=char(65+rand(3))	=B4/${B5}1/C4	/用随机符号生成一道题目
6		>C5=C5/"="/eval(C5)	>A2.write@a(C5)	/记录答案，输出一行

输出结果如下：

```
19+68=87
18+23=41
96-96=0
51-76=-25
84-79=5
63-12=51
73+34=107
45*80=3600
28*92=2576
62-10=52
```

注意

- **子程序部分不会自行执行**，只有在被调用时才会执行。上面例子中，前3行是主程序部分，在第3行中使用循环多次调用子程序。
- 子程序**可以置于网格中的任何位置**，而子程序主格的**代码块**就是子程序的范围。
- 在B2中需要先向文本文件中输出一次空字符，以清空文件中可能已有的数据。本例中的子程序只向文件中输出数据，而**无返回值**。

子程序中的参数

- SPL中的子程序类似于**自定义函数**，在调用时也可以向子程序输入参数。
- 调用子程序时，通过**func(C, x_1, x_2, ...)**的格式向子程序输入参数**x_1, x_2, ...**。
- 当子程序被输入参数时，如果只有1个参数，在子程序执行时，**参数值会赋值给子程序的主格**，有多个参数时，从第2个参数起会依次赋值给子程序主格**右侧的单元格。**

	A	B	C
1	=func(A2, 4, 6)	=func(A2, 23, 32)	/计算2个数的平均数
2	func		/子程序
3		return (A2+B2)/2	/返回2个数的平均数

A1,B1调用子程序计算后的结果如下：

Value
5.0

Value
27.5

注意

- A2中子程序每次被调用时，输入的2个参数依次赋值给A2和B2，B3中子程序返回的值可以用这两个格值执行所需计算。
- 如果子程序有输入参数，那么相关的单元格中不应再写入代码，子程序的代码可以从**下一行的代码块**区域开始。

变量的作用范围

子程序中变量的作用范围

- 主程序中的**单元格变量**或者执行了赋值的**变量**，在整个程序执行时都有效，如果子程序作了修改，这些变量的**值会变化**。
- 子程序中每次输入的**参数**，只在**子程序执行时有效**，子程序执行结束时，相关单元格变量会被清空。
- 子程序中，主格的代码块中，**其它单元格变量**的值，同样在**子程序执行时有效**。如果代码块中有常数单元格，每次子程序执行时都会**重新设入格值**。

	A	B	C	D
1	func	0	>B1=B1+1	/子程序
2		>A3=A3+A1	return A3/B1	/准备返回数据均值
3	0			/A3存储累积和
4	=func(A1,1)	=func(A1,2)	=func(A1,3)	/调用3次子程序

A4,B4,C4调用子程序计算后的结果如下：

Value
1.0

Value
3.0

Value
6.0

注意

- 本例中，子程序写在前两行中，子程序**仅在调用时才会执行**，代码位置和执行效果无关。
- 子程序需1个输入参数，每次执行子程序时，A3会计算参数的总和，而每次也都会将B1中的数值加1，并返回参数和/B1的结果。但是，从结果可以看出，子函数返回的就只是输入参数的累计和而已。这是由于每次执行子程序时，B1均会被赋值为输入的常数值0。

子程序中变量的作用范围

- 主程序中的**单元格变量**或者执行了赋值的**变量**，在整个程序执行时都有效，如果子程序作了修改，这些变量的**值会变化**。
- 子程序中每次输入的**参数**，只在**子程序执行时有效**，子程序执行结束时，相关单元格变量会被清空。
- 子程序中，主格的代码块中，**其它单元格变量**的值，同样在**子程序执行时有效**。如果代码块中有常数单元格，每次子程序执行时都会**重新设入格值**。

	A	B	C	D
1	func	0	>B3=B3+1	/子程序
2		>A3=A3+A1	return A3/B3	/准备返回数据均值
3	0	0		/A3存储累积和
4	=func(A1,1)	=func(A1,2)	=func(A1,3)	/调用3次子程序

A4,B4,C4调用子程序计算后的结果如下：

Value
1.0

Value
1.5

Value
2.0

注意

- 如果C1中计算数据个数也使用主程序中的单元格变量，在C2中返回的数据均值就正确了，三次累计均值分别为1.0， 1.5和2.0。

	A	B	C	D
1	func	0	>B3=B3+1	/子程序
2		>A3=A3+A1	return A3/B3	/准备返回数据均值
3	0	0		/A3存储累积和
4	=func(A1,1)	=func(A1,2)	=func(A1,3)	/调用3次子程序

我们以刚刚使用的子程序为例，了解一下子程序的调用过程：

这个程序中，前两行是子程序，**只有调用时才会执行**，否则不会执行到。所以，主程序**从第3行开始执行**，A3和B3都被赋值为0。

执行**A4**时，需要调用A1中的子程序，子程序有一个参数值为1。此时会跳到**A1**格，并将参数值1赋值给A1。接下来执行**子程序主格A1的代码块**，先执行第1行，**B1**被赋值为初始值0（这里实际上B1的值在子程序中并未使用）；执行**C1**，主程序中的单元格参数B3加1，记录子程序**执行次数**。接着执行**B2**，主程序中的单元格参数A3加上了A1中输入的参数，记录参数的**累计值**。**C2**中返回**累计平均值**1/1=1.0 ，子程序调用结束，子程序的结果返回给A4。

接着执行**B4**，再次调用A1中的子程序，到A1中执行子程序，参数值2赋值给A1，同样B1赋值为0，C1中累加执行次数到B3，B2中累加总和到A3，C2返回累计平均值3/2=1.5，子程序结束，结果返回给B4。

最后执行**C4**，又一次调用A1中的子程序，参数值3赋值给A1，B1赋值为常数值0，C1中累加执行次数到B3，B2中累加综合到A3，C2返回累计平均值6/3=2.0，子程序结束，结果返回给C4。

程序执行结束时，子程序所在的前两行格值均为空，A3的格值为累计和6，B3的格值为累计次数3。

	A	B	C	D
1	=func(A2, …)			/主程序调用子程序1
2	func	…	…	/子程序1
3		=func(A5, …)	…	/调用子程序2
4		…	return …	/子程序1返回
5	func	…	…	/子程序2
6		…	return …	/子程序2返回

子程序的嵌套

- SPL中，子程序中也可以调用其它子程序，在子程序中调用其它子程序的使用方法，叫做**子程序的嵌套**。
- 在子程序嵌套调用时，在调用嵌套子程序**结束后，会返回调用处**，继续执行原子程序，和主程序中调用子程序的执行顺序是类似的。

【例7.4】用嵌套子程序，输出九九乘法表到multable.txt。其中第n行输出1*n~n*n的算式。

7.3 子程序

【例7.4】用嵌套子程序，输出九九乘法表到multable.txt。其中第n行输出1*n~n*n的算式。

	A	B	C	D
1	9	=file("multable.txt")	>B1.write("")	/乘法表尺寸，初始化输出文件
2	for A1	>func(A5, A2)		/循环9次输出各行算式
3	func		=A3*B3	/子程序2，输入两个乘数，计算积
4		=A3/"*"/B3/"="/C3/"\t"	return B4	/生成一句口诀返回
5	func			/子程序1，输出一行口诀
6		for A5	>B5=B5+func(A3, B6, A5)	/循环子程序2，在B5中生成一行口诀
7		>B1.write@a(B5)		/输出一行到文件

输出结果如下：

```
1*1=1
1*2=2   2*2=4
1*3=3   2*3=6   3*3=9
1*4=4   2*4=8   3*4=12  4*4=16
1*5=5   2*5=10  3*5=15  4*5=20  5*5=25
1*6=6   2*6=12  3*6=18  4*6=24  5*6=30  6*6=36
1*7=7   2*7=14  3*7=21  4*7=28  5*7=35  6*7=42  7*7=49
1*8=8   2*8=16  3*8=24  4*8=32  5*8=40  6*8=48  7*8=56  8*8=64
1*9=9   2*9=18  3*9=27  4*9=36  5*9=45  6*9=54  7*9=63  8*9=72  9*9=81
```

注意

- 子程序1和子程序2在代码中的**位置是不影响使用的。**
- 在子程序1中，循环调用子程序2，每次生成一条口诀，从而得到整行口诀。通过使用**子程序**，可以使得主程序结构**简单清晰**，更容易理解。

子程序的递归

- SPL中，子程序可以调用其它子程序，如果在子程序中调用自身，则成为子程序的递归。
- 使用子程序的递归时要确保递归调用能够有效结束，**避免陷入死循环**。
- 子程序递归调用时，各个执行进程互不影响，子程序的**参数不会互相干扰**，只在各自执行范围内使用。

	A	B	C	D
1	func	if A1<=1	return 1	/子程序，n不大于1时设置阶乘为1
2		=A1*func(A1,A1-1)	return B2	/递归，n!=n*((n-1)!)
3	=func(A1,4)	=func(A1,7)	=func(A1,10)	/调用3次子程序

A3,B3,C3调用子程序计算后的结果如下：

Value
24

Value
5040

Value
3628800

注意

- 在递归调用子程序时，A1中的子程序被多次调用，主格A1中也被各自输入了不同的参数值，但是，子程序的运行是**各自独立的**，A1的值只在各自子程序的调用过程中有效，**不会互相干扰**。

7.4 跨网格调用

7.4 跨网格调用

跨网格调用

- SPL中，除了可以调用子程序之外，还可以跨网格调用，直接调用一个**dfx程序文件**，执行其中所有代码后返回。
- 类似于子程序，在跨网格调用时，被调用的网格如果需要返回结果，可以用**return**语句返回。
- 跨网格调用需要在主程序中使用**call(fn, …)**函数，fn是dfx文件名。

网格文件exer.dfx，随机生成一道两位数口算题：

	A	B	C
1	+	-	*
2	=rand(90)+10	=rand(90)+10	=char(65+rand(3))
3	=A2/${C2}1/B2	>A3=A3/"="/eval(A3)	return A3

在主网格中调用上面的网格文件：

	A	B	C
1	for 10	>output(call("exer.dfx"))	/循环执行网格文件

注意

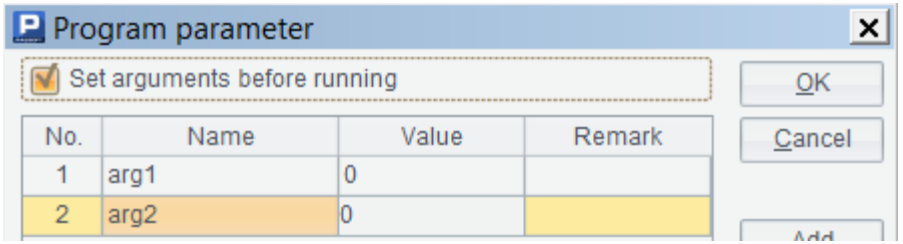
- 跨网格调用，和子函数的调用基本一致，只是子函数使用**func(C,…)**函数调用是需要指明子函数主格C，而跨网格调用函数**call(fn,…)**需要指明dfx文件名。
- 跨网格调用比子函数的通用性更高，**可以在不同的网格代码中调用**；同时安全性也更高，可以在不影响调用的前提下**加密**调用文件的代码。

7.4 跨网格调用

跨网格调用参数的传递

- 和子函数调用类似，跨网格调用时也可以使用参数，参数在调用函数call(fn, ...)的dfx文件名后输入。
- 如果被调用的网格需要使用参数，那么需要在网格中定义网格参数。
- 在SPL中定义网格参数，可以在集算器的菜单栏中点击Program>Parameter设置。

网格文件bigger.dfx，返回两个数之中较大的，需要先输入两个网格参数：



	A	B
1	if arg1>arg2	return arg1
2	else	return arg2

在主网格中调用上面的网格文件：

	A	B
1	=call("bigger.dfx", 72, 75)	/调用网格文件，求较大值

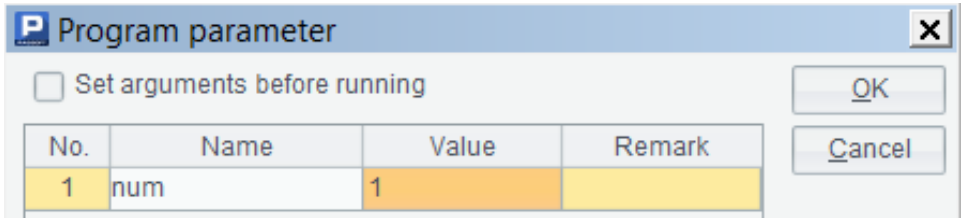
注意

- 调用带参数的网格文件时，不必写参数名，参数将按列表中的顺序依次设入。

【例7.5】用跨网格调用，输出九九乘法表到multable.txt。其中第n行输出1*n~n*n的算式。

【例7.5】用跨网格调用，输出九九乘法表到multable.txt。其中第n行输出1*n~n*n的算式。

网格文件mulline.dfx，输入参数num，输出一行乘法口诀，从1*num到num*num：



	A	B	C	D
1	func		=A1*B1	/子程序，输入两个乘数，计算积
2		=A1/"*"/B1/"="/C1/"\t"	return B2	/生成一句口诀返回
3	=""	for num	>A3=A3+func(A1, B3, num)	/循环子程序，在A3中生成口诀
4	return A3			/返回整行

在主程序中，循环调用网格文件，在文件中输出全部的乘法表：

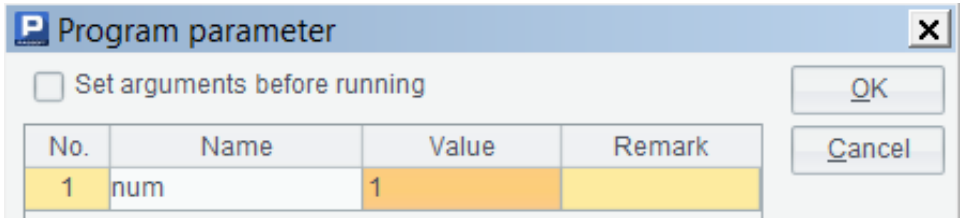
	A	B	C	D
1	9	=file("multable.txt")	>B1.write("")	/乘法表尺寸，初始化输出文件
2	for A1	=call("mulline.dfx", A2)	>B1.write@a(B2)	/循环9次输出各行算式

输出结果和前面是相同的。

跨网格调用的嵌套和递归

- 在前面的例子中，在被调用的网格中使用了子程序。实际上，被调用的网格中除了需要要用return语句返回所需值外，是没有任何限制的，**既可以使用子程序，也可以调用其它的网格。**
- 在跨网格调用的dfx中，又调用其它网格，就是**跨网格调用的嵌套**；如果在网格中调用自身，就是**跨网格调用的递归**。
- 被调用的网格代码在计算时**各自独立**，不会互相影响。

网格文件factorial.dfx，计算num的阶乘：



	A	B
1	if num<=1	return 1
2	else	return num*call("factorial.dfx", num-1)

在主网格中调用上面的网格文件：

	A	B
1	=call("factorial.dfx", 5)	/调用网格文件，计算阶乘

7.5 习题

7.5 习 题

【习题7.1】输入自己的生日，分别计算在此之后50年，500个月，5000天的日期（使用elapse函数，选项和interval类似）。

【习题7.2】调用子程序，输出斐波那契数列的前20项。

【习题7.3】递归调用子程序，计算a的n次方，其中n是正整数。

【习题7.4】嵌套调用子程序，输出所有的五角星数，五角星数是指这样的五位数：它各位数字的5次方的和，等于它本身。

【习题7.5】调用子程序，在calendar.txt中输出2020年全年的月历。在月历中，每周占一行，从周日开始显示，第一行显示英文月份，第二行显示星期缩写。2020年1月1日为周三。

【习题7.6】用跨网格调用，解决上一道题目，输出2020年全年的月历。

7.5 习 题

【习题7.1】输入自己的生日，分别计算在此之后50年，500个月，5000天的日期（使用elapse函数，选项和interval类似）。

7.5 习 题

【习题7.1】输入自己的生日，分别计算在此之后50年，500个月，5000天的日期（使用elapse函数，选项和interval类似）。

	A	B
1	=date(1992,6,23)	/输入生日日期
2	>output(elapse@y(A1,50))	/输出50年后日期
3	>output(elapse@m(A1,500))	/输出500个月后日期
4	>output(elapse(A1,5000))	/输出5000天后日期

结果：

2042-06-23
2034-02-23
2006-03-02

【习题7.2】调用子程序，输出斐波那契数列的前20项。

7.5 习 题

【习题7.2】调用子程序，输出斐波那契数列的前20项。

	A	B	C	D
1	20			/输入所需项数
2	for A1	>output(func(A3, A2))		/循环调用子程序，输出结果
3	func	if A3<3	return 1	/子程序，前两项是1
4		else	return func(A3, A3-1)+func(A3,A3-2)	/从第3项起，为前两项的和

结果：

```
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765
```

【习题7.3】递归调用子程序，计算a的n次方，其中n是正整数。

7.5 习 题

【习题7.3】递归调用子程序，计算a的n次方，其中n是正整数。

	A	B	C	D
1	1.5	3	>output(func(A2, A1, B1))	/输入a和n，调用子程序计算
2	func			/子程序，首行空出接收参数
3		if B2<=1	return A2	/n为1时，返回a
4		else	return A1*func(A2, A2, B2-1)	/n大于1时，返回A1*f(n-1)

结果：

3.375

【习题7.4】嵌套调用子程序，输出所有的五角星数，五角星数是指这样的五位数：它各位数字的5次方的和，等于它本身。

7.5 习 题

【习题7.4】嵌套调用子程序，输出所有的五角星数，五角星数是指这样的五位数：它各位数字的5次方的和，等于它本身。

	A	B	C	D
1	for 10000, 99999	if func(A2, A1)	>output(A1)	/循环所有5位数，调用子程序1判断是否五角星数
2	func	=string(A2)		/子程序1，将5位数转成字符串
3		=func(A6,left(B2, 1))	=func(A6,mid(B2, 2, 1))	/从字符串中取出万位和千位数字，调用子函数2
4		=func(A6,mid(B2, 3, 1))	=func(A6,mid(B2, 4, 1))	/从字符串中取出百位和十位数字
5		=func(A6,right(B2, 1))	return B3+C3+B4+C4+B5==A2	/取出个位数字，判断是否五角星数
6	func	=int(A6)	return B6*B6*B6*B6*B6	/子程序2，根据一位数字字符，计算它的5次方

结果：

```
54748
92727
93084
```

7.5 习 题

【习题7.5】调用子程序，在calendar.txt中输出2020年全年的月历。在月历中，每周占一行，从周日开始显示，第一行显示英文月份，第二行显示星期缩写。2020年1月1日为周三。

7.5 习 题

【习题7.5】调用子程序，在calendar.txt中输出2020年全年的月历。在月历中，每周占一行，从周日开始显示，第一行显示英文月份，第二行显示星期缩写。2020年1月1日为周三。

	A	B	C	D	E
1	=date(2020,1,1	=A1-pdate(A1)	=file("calendar.txt")	>C1.write("2020")	/计算星期的初始值，初始化文件
2	for 7	=pdate(A1)+A2-1	>D1=D1/string(B2, "E")/"\t"		/准备星期缩写标题行，记录在D1中
3	for 12	>func(A4, A3)			/调用子程序，输出各月月历
4	func	=date(2020,A4,1)	=days(B4)	=""	/子程序，输出一月月历，计算当月天数
5		>C1.write@a(string(B4, "MMMM"))	>C1.write@a(D1)	>D4=pad("","\t",B1%7)	/输出月份和星期标题
6		for C4	>D4=D4/B6/"\t"	>B1=B1%7+1	/添加本日日期，设置明日的星期
7			if B1==7 || B6==C4	>C1.write@a(D4),D4=""	/周末或月底，输出一周记录

结果：

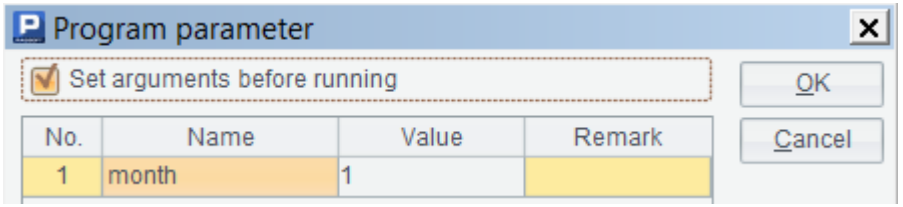
22 23 24 25 26 27 28
29 30
December
Sun Mon Tue Wed Thu Fri Sat
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

【习题7.6】用跨网格调用，解决上一道题目，输出2020年全年的月历。

7.5 习 题

【习题7.6】用跨网格调用，解决上一道题目，输出2020年全年的月历。

网格文件mcalendar.dfx，输入参数month，输出一个月的月历：



	A	B	C	D	E
1	=date(2020, month,1)	=A1-pdate(A1)	=days(A1)	=string(A1, "MMMM")+"\r\n"	/计算本月首日的星期，C1中初始化本月月历
2	for 7	=pdate(A1)+A2-1	>D1=D1/string(B2, "E")/"\t"		/星期标题行
3	=pad("","\t",B1%7)	for C1	>A3=A3/B3/"\t"	>B1=B1%7+1	/初始化首周空位，循环本月天数
4			if B1==7 || B3==C1	>D1=D1/"\r\n"/A3,A3=""	/周末或月底，添加一周的月历
5	return D1				/返回本月月历

在主程序中，循环调用网格文件，在文件中输出全年月历：

	A	B	C
1	=file("calendar.txt")	>A1.write("2020")	/初始化输出文件
2	for 12	>A1.write@a(call("mcalendar.dfx", A2))	/循环跨网格调用，输出每月月历

输出结果和前面是相同的，注意文本文件中需要输出多行数据时，换行符是\r\n。

第8章 序列

8.1 序列的概念

8.2 序列的产生与修改

- 序列常数
- 序列的成员访问
- 在序列最后添加成员

8.3 序列的循环与计算

- 用 for A 执行循环
- 聚合
- 序列计算举例

8.4 常见的序列

- 数列
- 字符串序列

8.5 多维序列

- 什么是多维序列
- 多维序列成员的访问
- 多维序列的使用

8.6 习题

8.1 序列的概念

8.1 序列的概念

为什么需要序列?

- 在解决问题时，经常需要处理许多**用途相同，意义相同**的数据。要简洁而高效地处理这些数据，就需要**序列**。
- 使用序列，可以方便地实现数据的存储与读取，也可以方便地实现**循环计算**。
- 序列中的数据，还可以进行**集合间运算**。或调用一些特殊功能**函数**，实现许多常见的数据处理方法，如求和、排序等。

8.1 序列的概念

假设我们去超市买了10件商品，参考前几章的内容，如果我们想算总价，可能需要这样处理：

	A	B	C	D	E	F
1	23.5	29.1	15.8	4.6	5.2	/原始数据
2	5.2	2.5	7.6	12.9	43.8	/原始数据
3	=A1+B1+C1+D1+E1+A2+B2+C2+D2+E2					/直接求和法

A3	Value
	150.2

显然这样的写法既繁琐又浪费空间。如果数据量进一步扩大，所占用的空间和计算表达式的繁琐程度将会是令人难以接受的。

8.1 序列的概念

若使用序列及相关的特殊功能函数，就可以实现上面例题中计算的简化。

	A	B
1	[23.5, 29.1, 15.8, 4.6, 5.2, 5.2, 2.5, 7.6, 12.9, 43.8]	/将数据保存在序列中
2	=A1.sum()	/使用聚合函数求和

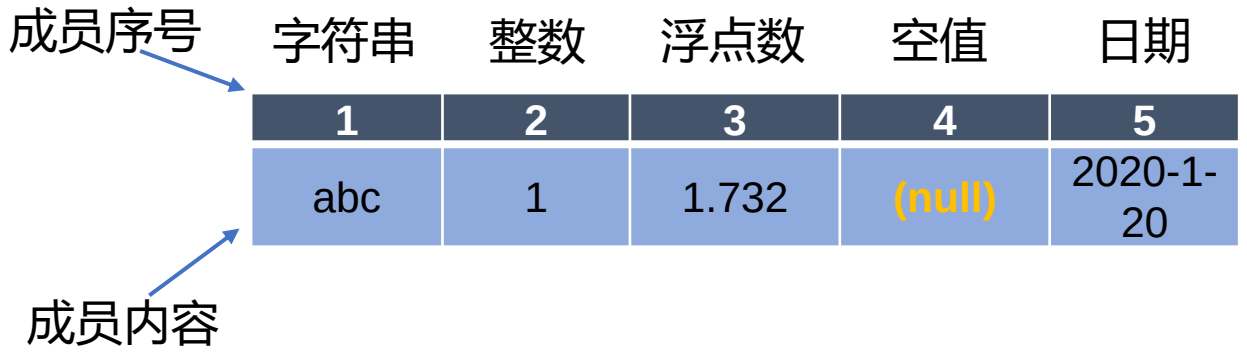
A2	Value
	150.2

既然例题中谈到了使用序列，那么序列到底是什么呢?

8.1 序列的概念

序列的概念

- **序列**，是一组**有顺序**的数据的**集合**。组成序列的数据，被称为序列的**成员**。序列中的成员可以是**任意**数据类型，比如字符串、数字、浮点、日期，还可以为**空值**。序列中的成员的值可以**重复**。
- 序列类似于某些程序语言中的**数组**。



注意

- **(null)**代表空值。

8.1 序列的概念

因为序列是有序的，我们可以利用它的有序性，处理许多实际问题。比如：可以使用序列，存储1个5人小组的学生们的数学成绩。序列中数据的顺序，即为学生们学号的顺序。

5人小组的数学成绩

1	2	3	4	5
91	87	65	23	59

假如需要查询学号为3的学生的数学成绩，只需找出序号为3的序列成员的值：65。

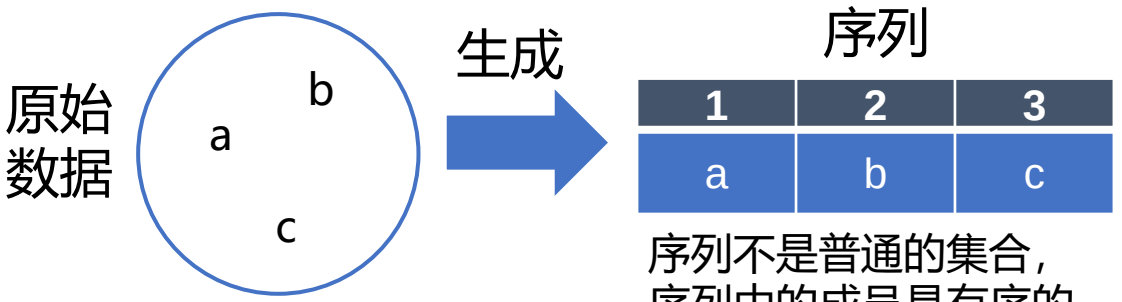
下面我们将介绍序列的生成，读写，循环，常见类型，以及多维序列的概念及使用。

8.2 序列的产生与修改

8.2.1 序列常数

序列的生成

- 常用的有常数构造和函数构造两种方法，也可以通过计算生成。这里我们主要介绍常数构造法。



序列不是普通的集合，序列中的成员是有序的。

常数构造法

	A	B
1	[a, b, c]	/在SPL中用常数构造法生成序列

点击A1，在右侧的值显示区，可以查看序列：

A1	Index	Member
	1	a
	2	b
	3	c

显示时，Index一列是辅助列，用来显示序号。序列中成员的总个数，称为序列的长度。

8.2.1 序列常数

序列的特殊类型

- 数列指成员全是整数的序列。
- 空列指没有成员的序列，注意空列和空值null是不同的。

数列：成员全是整数

A1	Index	Member
	1	1
	2	2
	3	3

	A	B
1	[1,2,3]	/用常数构造法定义数列

空列：没有成员

A1	Index	Member

空列是个没有成员的序列；
而空值会显示为(null)

	A	B
1	[]	/用常数构造法定义空列

8.2.1 序列常数

除了直接设定数据成员外，也可以用已存在的**单元格值**作为序列成员创建新序列。

	A	B	C
1	s	a	/原始数据
2	d	t	/原始数据
3	=[A1:A2]	=[A1:B2]	/用格值定义序列

A3	Index	Member
	1	s
	2	d

B3	Index	Member
	1	s
	2	a
	3	d
	4	t

注意

- 当原始数据占据多行多列，比如写作 [A1:Bn]时，SPL会按先行后列的顺序将数据依次放入序列中，即：[A1, B1, A2, B2, ..., An, Bn]。

8.2.1 序列常数

还可以使用已有的**序列做乘法计算**生成新的序列。

	A	B
1	=["a"]*4	/序列成员的数据类型为普通数据时，用乘法构成的新序列中，所有成员都复制序列成员的数据
2	[0]	
3	=[A2]*4	/当A2本身为序列时，[A2]表示序列的序列。此时用乘法[A2]*4构成的新序列中，所有成员都引用A2中的序列

A1	Index	Member
	1	a
	2	a
	3	a
	4	a

A3	Index	Member
	1	[0]
	2	[0]
	3	[0]
	4	[0]

注意

- 在A1中，[“a”]这样的序列常数**直接用在表达式中**，当序列常数直接用在表达式中时，序列中的成员**解析规则与常数格不同**，字符串类型的成员需要**用引号括起来**，时间日期类型的字符串需要**先行转换**。
- 对于[A]*n，当A为序列时，生成的序列B中所有的成员实际上都是引用的**同一个对象**，如果此时修改了A， B中的各个成员都会被修改。

8.2.1 序列常数

【例8.1】假设某医院有一个简易的排队叫号系统，试用序列模拟这一排队叫号系统。假设序列成员的先后顺序即患者排队的顺序。现有5名患者在排队中，依序为：James、Kate、Bill、Mary、Lucy。

8.2.1 序列常数

【例8.1】假设某医院有一个简易的排队叫号系统，试用序列模拟这一排队叫号系统。假设序列成员的先后顺序即患者排队的顺序。现有5名患者在排队中，依序为：James、Kate、Bill、Mary、Lucy。

解题思路

让序列中的成员**顺序**与实际中患者排号顺序相一致即可。

	A	B
1	[James, Kate, Bill, Mary, Lucy]	/用单元格常数构建序列，常数序列可以直接显示
2	=["James", "Kate", "Bill", "Mary", "Lucy"]	/用表达式构建序列，表达式中序列的字符串成员要用引号表示，表达式需要计算才会有结果，结果和A1相同

A1	Index	Member
	1	James
	2	Kate
	3	Bill
	4	Mary
	5	Lucy

A2	Index	Member
	1	James
	2	Kate
	3	Bill
	4	Mary
	5	Lucy

8.2.2 序列的成员访问

序列成员的访问

- 最基本的访问序列成员的方式是在表达式中用**A(n)**访问，根据序号**n**，访问**A**中指定位置的成员。除了获取指定位置的成员，也可以用这种访问方式为序列成员赋值：**A(n)=x**，但是只能用这种方式改变序列中**已有成员**的值，**不能增加序列成员**，即n不能超过序列范围。
- 还可以用函数**A.m(n)**的方式访问序列**A**的成员，参数中**n**可以是成员的序号，也可以用负数表示倒数第几个位置的成员。

A(2)：第2个成员

A(4)：第4个成员

A.m(3)：第3个成员

A.m(-2)：倒数第2个成员

Index	Member
1	a
2	b
3	c
4	d
5	e

8.2.2 序列的成员访问

【例8.2】假设例8.1中，第3位患者Bill即将问诊结束，医院准备通过广播提醒排在他后面的下一位患者做好准备，那么该如何获取下一位患者的名字？如果医院为了安抚排名靠后的患者，准备给队列中的最后一名患者一个小礼物，该怎样快速找到排在最后一名的患者？

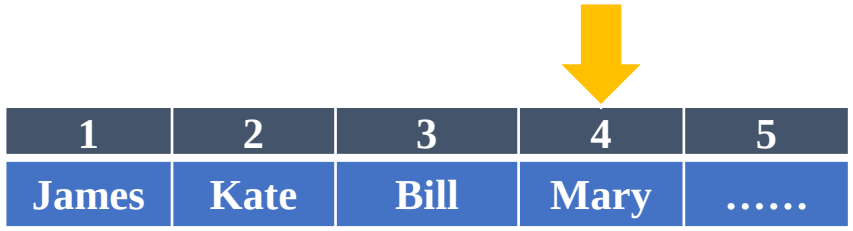
8.2.2 序列的成员访问

【例8.2】假设例8.1中，第3位患者Bill即将问诊结束，医院准备通过广播提醒排在他后面的下一位患者做好准备，那么该如何获取下一位患者的名字？如果医院为了安抚排名靠后的患者，准备给队列中的最后一名患者一个小礼物，该怎样快速找到排在最后一名的患者？

解题思路

可以通过**序号**来访问序列的成员。对于靠后的序列成员，还可以使用序列成员的**相对位置**访问法快速找到。

SPL的表达式中，可以用 **A(i) 按序号 i 访问**序列A中的成员，**i 只能为正数**，计算时如果 **i 超出A的序号范围**会**报错**。还可以用函数 **A.m(i) 反向访问**序列A中的成员，当 **i** 为负数时代表从序列的末尾开始，**反向**访问第 **-i** 个成员；计算时如果 **i 超出A的序号范围**会**返回空值**。



序列中第4个成员的值：Mary

序列中最后1名成员的值：Lucy

8.2.2 序列的成员访问

【例8.2】假设例8.1中，第3位患者Bill即将问诊结束，医院准备通过广播提醒排在他后面的下一位患者做好准备，那么该如何获取下一位患者的名字？如果医院为了安抚排名靠后的患者，准备给队列中的最后一名患者一个小礼物，该怎样快速找到排在最后一名的患者？

	A	B
1	[James, Kate, Bill, Mary, Lucy]	/常数构造法定义序列A1
2	=A1(4)	/取序列内第4个成员对应的值：Mary
3	=A1.m(4)	/同A2
4	=A1.m(-1)	/反向访问序列内倒数第1个成员对应的值：Lucy

A2	Value
	Mary

A3	Value
	Mary

A4	Value
	Lucy

8.2.2 序列的成员访问

【例8.3】假设例8.1中，医院将第4名患者的名字：Marie，错误地登记为：Mary。怎样快速地将该患者的名字修正正确?

8.2.2 序列的成员访问

【例8.3】假设例8.1中，医院将第4名患者的名字：Marie，错误地登记为：Mary。怎样快速地将该患者的名字修正正确?

解题思路

可以通过对指定位置的序列成员**直接赋值**的方法。

SPL中，表达式可以写成**A(i)=x**的形式。其中**A**代表序列，**i**代表要访问的成员的序号，**x**代表要赋给该成员的新值。

8.2.2 序列的成员访问

【例8.3】假设例8.1中，医院将第4名患者的名字：Marie，错误地登记为：Mary。怎样快速地将该患者的名字修正正确?

	A	B
1	[James, Kate, Bill, Mary, Lucy]	/常数构造法定义序列A1
2	>A1(4)="Marie"	/将序列内第4个成员对应的值修改为：Marie

执行后，A1中序列被修改：

A1	Index	Member
	1	James
	2	Kate
	3	Bill
	4	Marie
	5	Lucy

8.2.2 序列的成员访问

【例8.4】假设例8.1中，医院将第4名患者与第5名患者写错了顺序，怎样快速地修正?

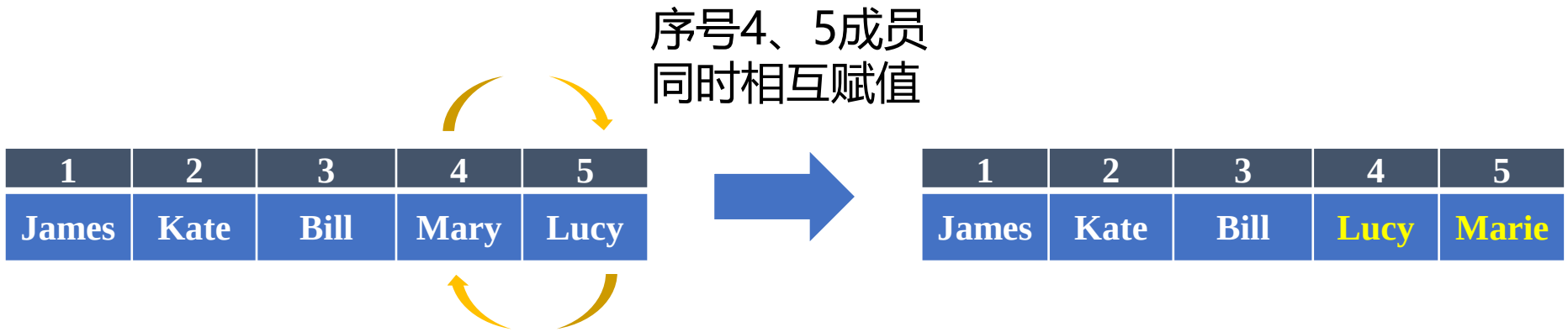
8.2.2 序列的成员访问

【例8.4】假设例8.1中，医院将第4名患者与第5名患者写错了顺序，怎样快速地修正?

解题思路

可以通过对多个被引用的序列成员同时进行赋值的方法。

SPL中，可以用函数A(p)=X的形式为序列A中的多个成员赋值，其中p为序号**数列**，X为序列。这表示将数列p对应的序列A成员依次赋值为序列X中的数据，其中p和X的**序列长度要相等**，且p中的各个序号不能超过A的范围。



8.2.2 序列的成员访问

【例8.4】假设例8.1中，医院将第4名患者与第5名患者写错了顺序，怎样快速地修正?

	A	B
1	[James, Kate, Bill, Mary, Lucy]	/常数构造法定义序列A1
2	>A1([4,5])=A1([5,4])	/将序列内第4个与第5个成员的值相互对调

执行后，A1中序列被修改：

A1	Index	Member
	1	James
	2	Kate
	3	Bill
	4	Lucy
	5	Mary

8.2.3 在序列最后添加成员

在序列最后添加成员

- 在序列A**最后**添加成员，可以用表达式**>A=A|x**。这种方式实际上是重新为A赋值，而并不是直接改变原序列本身。
- 有关序列的修改和维护，将在下一章详细介绍。

	A	B
1	[a,b,c]	/用常数构造法定义数列
2	>A1=A1|"new"	/更新A1，在序列最后添加一个成员

A1	Index	Member
	1	a
	2	b
	3	c

添加成员

A1	Index	Member
	1	a
	2	b
	3	c
	4	new

8.2.3 在序列最后添加成员

【例8.5】假设例8.1中，又陆续新来了两位患者Ella和David，请列出他们加入后候诊队伍。

8.2.3 在序列最后添加成员

【例8.5】假设例8.1中，又陆续新来了两位患者Ella和David，请列出他们加入后候诊队伍。

解题思路

在序列的最后依次加入成员即可。为了关注序列中成员数的变化，在B列中计算了序列变化过程中序列长度的变化情况，用函数**A.len()**可以获得序列A中成员的总数，即序列A的**序列长度**。

	A	B	C
1	[James, Kate, Bill, Mary, Lucy]	=A1.len()	/用单元格常数构建序列，常数序列可以直接显示
2	>A1=A1|"Ella"	=A1.len()	/在序列最后添加一个成员Ella
3	>A1=A1|"David"	=A1.len()	/在序列最后添加一个成员David

A1	Index	Member
	1	James
	2	Kate
	3	Bill
	4	Mary
	5	Lucy

B1	Value
	5

A1	Index	Member
	1	James
	2	Kate
	3	Bill
	4	Mary
	5	Lucy
	6	Ella

B2	Value
	6

A1	Index	Member
	1	James
	2	Kate
	3	Bill
	4	Mary
	5	Lucy
	6	Ella
	7	David

B3	Value
	7

8.3 序列的循环与计算

8.3.1 用for A执行循环

for A循环

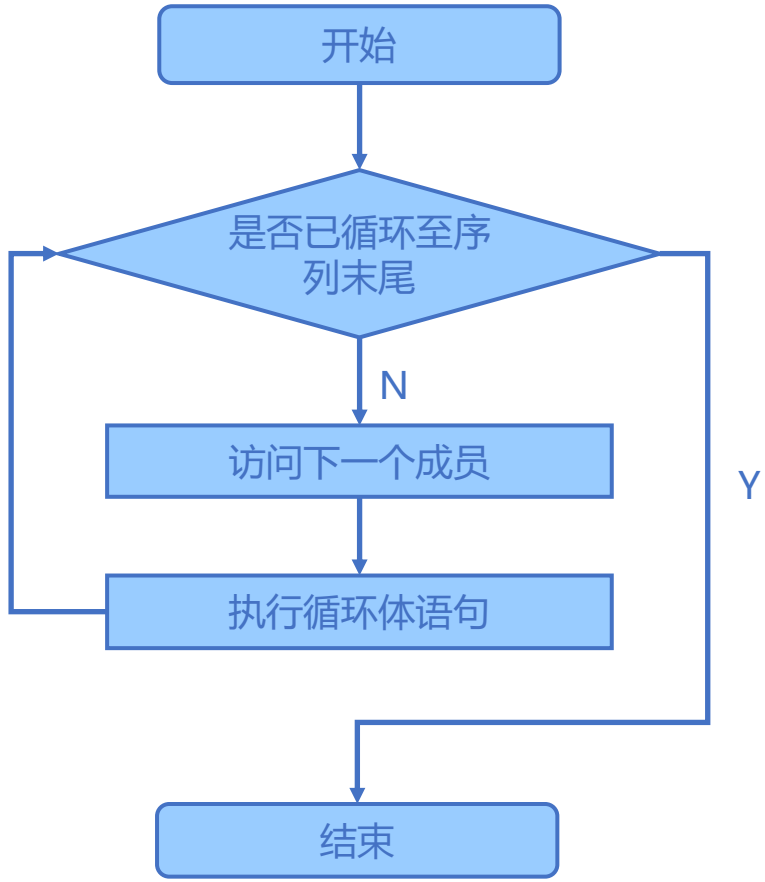
- 序列的for A循环，即针对序列中的**每一个成员**执行**重复性**的计算。

for A循环

	A	B
1	[a, b, c]	
2	for A1	>output(A2)

输出结果:

```
a
b
c
```



8.3.1 用for A执行循环

【例8.6】已知某研发小组所有组员出生日期：1978-04-25、1992-05-28、1983-06-16、1988-08-08、1992-05-17、1986-12-11，求所有成员的年龄。（按当前日期2020年6月6日计算）。

8.3.1 用for A执行循环

【例8.6】已知某研发小组所有组员出生日期：1978-04-25、1992-05-28、1983-06-16、1988-08-08、1992-05-17、1986-12-11，求所有成员的年龄。（按当前日期2020年6月6日计算）。

解题思路

对于这个问题，可以将研发小组所有组员的出生日期保存为一个序列，再使用针对序列的循环计算，依次计算出所有组员的年龄。

for A：循环序列A的成员，**主格**的值依次为当前A的成员。

	A	B	C	D
1	[1978-04-25,1992-05-28,1983-06-16,1988-08-08,1992-05-17,1986-12-11]	[]		/A1存储出生日期序列，B1中定义空序列准备存储结果
2	for A1	=age(A2)	>B1=B1|B2	/根据组员出生日期计算当前年龄并在B1中添加一个成员保存

B1	Index	Member
	1	42
	2	27
	3	36
	4	31
	5	28
	6	33

8.3.1 用for A执行循环

【例8.7】例8.6中，如果该小组组长年龄为33岁，并且没有同龄组员，并且组员在序列中的出生日期所对应的成员序号即为组员的工号。求该小组组长的工号。

8.3.1 用for A执行循环

【例8.7】例8.6中，如果该小组组长年龄为33岁，并且没有同龄组员，并且组员在序列中的出生日期所对应的成员序号即为组员的工号。求该小组组长的工号。

解题思路

要解答这个问题，必须在循环访问序列成员的同时，能够获得成员的序号。因为for A是依次循环，所以成员的序号就等于当前的循环次数。

#A：获得**当前**的**循环次数**，其中A为循环体的**主格**。

	A	B	C	D
1	[1978-04-25,1992-05-28,1983-06-16,1988-08-08,1992-05-17,1986-12-11	0		/A1存储出生日期序列，B1存储结果，初始化为0
2	for A1	=age(A2)		/根据组员出生日期计算当前年龄
3		if B2==33	>B1=#A2	/如果年龄为33岁则在B1中存储

B1	Value
	6

8.3.1 用for A执行循环

【例8.8】例8.6中，求小组内年龄最大的组员对应的工号。

8.3.1 用for A执行循环

【例8.8】例8.6中，求小组内年龄最大的组员对应的工号。

解题思路

解题时需要保存每次循环中的计算结果，并在下一次循环当中继续使用。用于保存循环结果的变量，应在**循环体外**定义，并注意设置初始值。

	A	B	C	D
1	[1978-04-25,1992-05-28,1983-06-16,1988-08-08,1992-05-17,1986-12-11]	0	0	/A1定义出生日期序列，B1存储最大年龄，C1存储对应工号
2	for A1	=age(A2)		/根据组员出生日期计算当前年龄
3		if B2>B1	>B1=B2	/B1记录该组最大年龄
4			>C1=#A2	/C1记录该组最大年龄组员的工号

C1	Value
	1

8.3.1 用for A执行循环

【例8.9】由2,3,6,9能够组成多少个各位数字全不同的3位数，它们是哪些数?

8.3.1 用for A执行循环

【例8.9】由2,3,6,9能够组成多少个各位数字全不同的3位数，它们是哪些数?

解题思路

题目中需要使用的数字**并不是连续的**，因此需要用一个**序列**存储，然后再循环序列计算。可以将结果存储在一个序列中，每次找到一个符合条件的3位数就将它添加到结果序列中，计数可以用结果**序列的长度**来计算。

	A	B	C	D	E	F
1	[2, 3, 6, 9]	[]				/准备数字序列，B1中定义空序列用来存储结果
2	for A1	for A1	for A1	if A2!=B2 && A2!=C2 && B2!=C2	>B1=B1|(A2*100+B2*10+C2)	/循环各位数字，找到一个各位不同的3位数则存储到B1中
3	=B1.len()					/输出总个数

B1	Index	Member
	1	236
	2	239
	3	263
	4	269
	5	293
	6	296
	7	326
	…	…

A3	Value
	24

8.3.2 聚合

聚合

- 在循环时，有时需要针对所有成员统计出**聚合**结果，如**计数、求和、求平均值，求极值**等。
- 用**for A**循环，可以通过循环序列中的每个成员，计算聚合结果。

计数：计算序列成员总个数

	A	B
1	[a, b, c]	0
2	for A1	>B1=B1+1

B1	Value
	3

8.3.2 聚合

【例题8.10】已知几名学生某次测验的分数依次如下：66, null, 82, 96，其中有人因故未参加测试因此无成绩，请计算他们的总分和平均分，缺考学生不参加统计。

8.3.2 聚合

【例题8.10】已知几名学生某次测验的分数依次如下：66, null, 82, 96，其中有人因故未参加测试因此无成绩，请计算他们的总分和平均分，缺考学生不参加统计。

解题思路

计算平均分，需要分别统计参加考试学生的总分和总人数。

	A	B	C	D
1	[66, null, 82, 96]	0	0	/生成分数序列，B1和C1分别存储总人数和总分
2	for A1	if A2 != null	>B1=B1+1	/如果有成绩，累计总人数
3			>C1=C1+A2	/累加总分
4	=round(C1/B1, 2)			/计算平均分，结果保留2位小数

B1	Value
	3

C1	Value
	244

A4	Value
	81.33

注意

- 有关聚合的计算，更方便的方法是使用**聚合函数**，有关聚合函数的使用，将在下一章中讲解。

8.3.3 序列的综合计算

【例8.11】已知某研发小组所有组员出生日期：1978-04-25、1992-05-28、1983-06-16、1988-08-08、1992-05-17、1986-12-11。如果序列的最后一位是组长的生日，请找出该小组内所有年龄比组长大的组员的工号。

8.3.3 序列的综合计算

【例8.11】已知某研发小组所有组员出生日期：1978-04-25、1992-05-28、1983-06-16、1988-08-08、1992-05-17、1986-12-11。如果序列的最后一位是组长的生日，请找出该小组内所有年龄比组长大的组员的工号。

解题思路

这里推荐使用**for n**形式的循环，每次循环的值就可以作为序号，获取序列中成员来计算年龄。这样可以方便地排除for A循环中组长与自己做对比的一次无意义的循环。

	A	B	C	D
1	[1978-04-25, 1992-05-28, 1983-06-16, 1988-08-08, 1992-05-17, 1986-12-11]	=age(A1.m(-1))	[]	/A1定义出生日期序列，B1算出组长年龄C1存储结果序列
2	for A1.len()-1	=age(A1(A2))		/根据组员出生日期计算当前年龄
3		if B2>B1	>C1=C1|#A2	/如果年龄大于组长则输出对应的工号

C1	Index	Member
	1	1
	2	3

注意

- 用序列循环计算时，可以根据需要选择**for A**还是**for n**，后者使用更灵活。
- 使用**for n**处理序列循环时往往需要从序列中按序号获取成员后再计算，此时要注意n的取值不能超过序列的长度，否则会报索引越界的错误。

8.3.3 序列的综合计算

【例8.12】假设有某班某科考试成绩：82，77，65，85，73，91，68，76，65。现想给这些考试成绩从高到低排个名次。

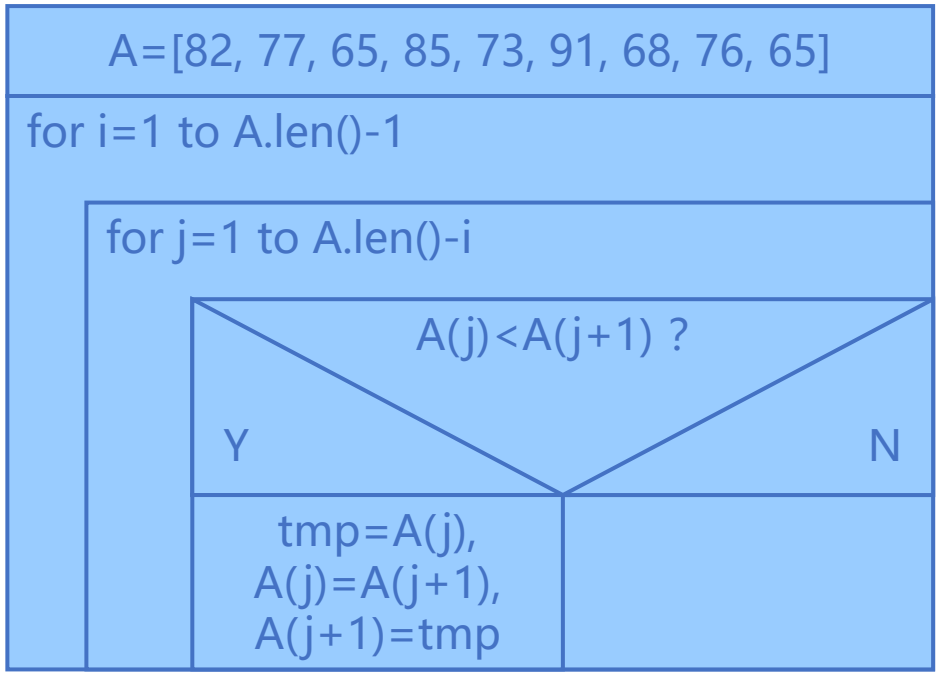
8.3.3 序列的综合计算

【例8.12】假设有某班某科考试成绩：82，77，65，85，73，91，68，76，65。现想给这些考试成绩从高到低排个名次。

解题思路

先将考试成绩放入序列中备用。循环比较每对**相邻**的考试成绩，如果前面的小于后面的，则交换二者位置。每一轮比较下来，都会将这一轮中的最低考试成绩放在**最后面**的位置。下一轮的比较终点设在上一轮比较所得到的最低考试成绩的**前一位**。

因为升序（降序）排列时， 较小（较大）的元素会经由交换，慢慢“浮”到序列**靠边缘**一端，就如同碳酸饮料中二氧化碳的气泡最终会上浮到液面一样，故此算法的名字为“**冒泡排序**”。



8.3.3 序列的综合计算

【例8.12】假设有某班某科考试成绩：82，77，65，85，73，91，68，76，65。现想给这些考试成绩从高到低排个名次。

	A	B	C	D	E
1	[82,77,65,85,73,91,68,76,65]				/原始数据存入序列中
2	for A1.len()-1	for A1.len()-A2	if A1(B2)<A1(B2+1)	>A1([B2, B2+1])=A1([B2+1, B2])	/循环比较每对相邻的值，如果前面的小于后面的，则交换位置

A1	Index	Member
	1	91
	2	85
	3	82
	4	77
	5	76
	6	73
	7	68
	8	65
	9	65

注意

- 本题也使用了**for n**（n为正整数）的方法处理序列的循环计算。
- B2中的n值会随A2值的改变而改变。

8.4 常见的序列

8.4.1 数列

数列

- 完全由**整数**成员组成的序列被称为**数列**。SPL中的数列更多的情况下会作为成员的序号列使用。
- 支持某些要求序列成员必须都是数值类型的特殊计算，比如求总和或求平均值等。
- 数列成员不会为空值(null)，所以序列长度A.len()等于求总数计算A.count()。有关求总数计数的计算将在下一章讲解。

数列

Index	Member
1	91
2	85
3	82
4	77
5	76
6	73
7	68
8	65
9	65

成员全是整数

8.4.1 数列

【例8.13】假设有5人学习小组的考试成绩：80，62，53，60，45。想计算此小组的平均分。

8.4.1 数列

【例8.13】假设有5人学习小组的考试成绩：80，62，53，60，45。想计算此小组的平均分。

解题思路

遍历序列中所有成员，通过累加算出总分，再除以总人数（即序列长度）即得平均分。

	A	B	C
1	[80,62,53,60,45]	0	/初始化序列，B1用来记录总分
2	for A1	>B1=B1+A2	/循环累加总分
3	=B1/A1.len()		/算出平均分

A3	Value
	60.0

【例8.14】假设例8.13中，考试达到60分记为合格，否则为不合格。求该小组的合格率。

8.4.1 数列

【例8.14】假设例8.13中，考试达到60分记为合格，否则为不合格。求该小组的合格率。

解题思路

遍历序列中所有成员，60分及以上的总计数加1，否则略过不计。最后除以总人数（即序列长度）即得合格率。

	A	B	C	D
1	[80,62,53,60,45]	0		/初始化序列，B1用来计数合格人数
2	for A1	if A2>=60	>B1=B1+1	/达到60分则合格数加1，否则不计
3	=B1/A1.len()	=string(A3,"#.0 %")		/算出合格率，在B3中显示成百分数

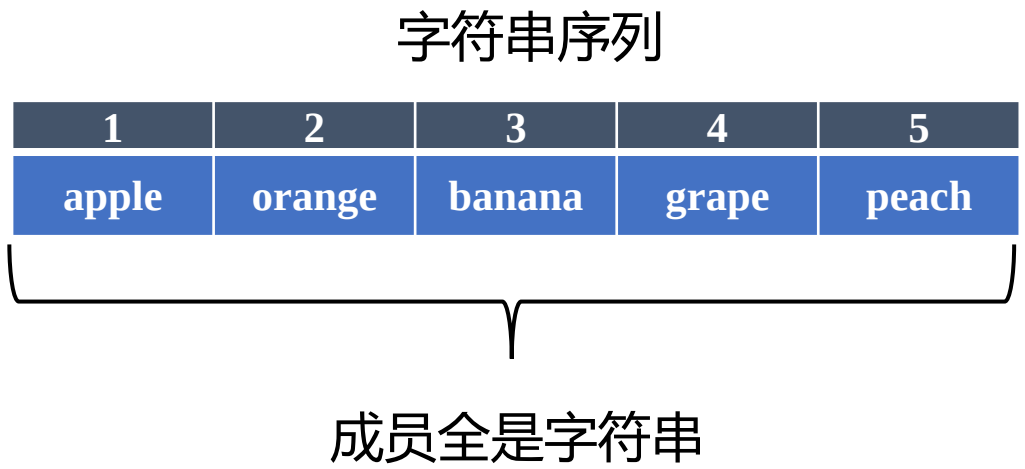
A3	Value
	0.6

B3	Value
	60.0 %

8.4.2 字符串序列

字符串序列

- 完全由字符串成员组成的序列即字符串序列。
- 因为字符串序列的成员都是字符串，所以可以调用字符串的专用函数进行一些特殊的计算，比如字符串的拆分与合并等。



8.4.2 字符串序列

【例8.15】有若干人名：Dennis，Rose，John，James，Robert，Jimmy。想输出首字母（默认大写）是 J 的人名。

8.4.2 字符串序列

【例8.15】有若干人名：Dennis，Rose，John，James，Robert，Jimmy。想输出首字母（默认大写）是 J 的人名。

解题思路

遍历序列中所有成员，截取首字母，判断是否为大写字母J。

	A	B	C	D
1	[Dennis, Rose, John, James, Robert, Jimmy]	[]		
2	for A1	if left(A2,1) == "J"	>B1=B1|A2	/首字母是 J 的人名存储入B1

B1	Index	Member
	1	John
	2	James
	3	Jimmy

【例8.16】假设例8.15中，想输出尾字母（默认小写）是 s 的人名。

8.4.2 字符串序列

【例8.16】假设例8.15中，想输出尾字母（默认小写）是 s 的人名。

解题思路

遍历序列中所有成员，截取尾字母，判断是否为小写字母s。

	A	B	C	D
1	[Dennis, Rose, John, James, Robert, Jimmy]	[]		
2	for A1	if right(A2,1) == "s"	>B1=B1|A2	/存储尾字母是 s 的人名

B1	Index	Member
	1	Dennis
	2	James

8.4.2 字符串序列

【例8.17】假设例8.15中，这几人的全名依次为：Dennis White，Rose Smith，John Davis，James Smith，Robert Jones，Jimmy Thomas。已知其中只有一位姓氏以D开头的，是其余几人的组长。试找出组长是哪一人。

8.4.2 字符串序列

【例8.17】假设例8.15中，这几人的全名依次为：Dennis White，Rose Smith，John Davis，James Smith，Robert Jones，Jimmy Thomas。已知其中只有一位姓氏以D开头的，是其余几人的组长。试找出组长是哪一人。

解题思路

遍历序列中所有成员，先以空格为分隔符截取后面的姓氏部分的字符串，再截取首字母，若为大写字母D，则存储人物全名并结束循环。

	A	B	C	D
1	[Dennis White, Rose Smith, John Davis, James Smith, Robert Jones, Jimmy Thomas]			
2	for A1	if mid(A2,pos@z(A2," ")+1,1)=="D"	>B1=A2	/存储首个姓氏以D开头的人名
3			break	/提前结束循环

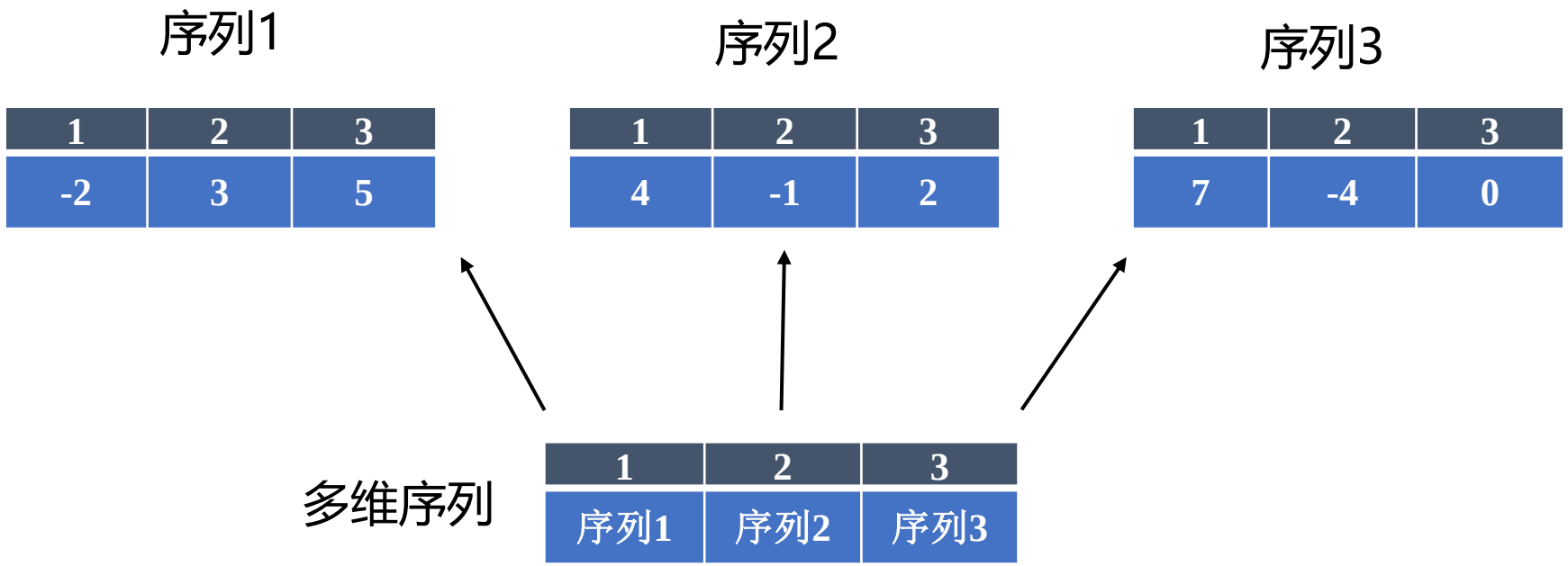
B1	Value
	John Davis

8.5 多维序列

8.5.1 什么是多维序列

多维序列

- 序列中的成员可以是**任何数据类型**。当一个**序列的成员**本身也是**序列**时，它就被称为**多维序列**。



8.5.1 什么是多维序列

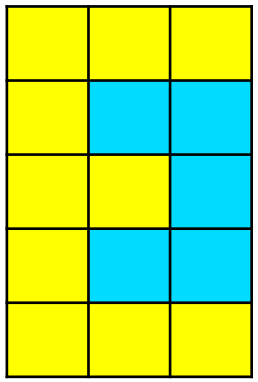
比如，矩阵就可视为一种多维序列。

使用SPL生成矩阵：$\begin{bmatrix} 11 & 23 \\ 59 & 17 \end{bmatrix}$

	A	B
1	[[11, 23] ,[59, 17]]	/使用常数构造法创建矩阵

实际生活中用于文字显示功能的点阵，也可以用多维序列显示。

用点阵显示文字：E



此点阵在SPL中可保存为：

	A
1	[[1,1,1],[1,0,0],[1,1,0],[1,0,0],[1,1,1]]

8.5.2 多维序列成员的访问

访问多维序列

- 多维序列的成员访问是**分级**的，先访问作为成员的序列，再访问成员序列里面的成员。

8.5.2 多维序列成员的访问

在SPL中逐级访问矩阵的元素

	A	B
1	[[11, 23] ,[59, 17]]	/常数构造法生成多维序列：$\begin{bmatrix}11 & 23\\59 & 17\end{bmatrix}$
2	=A1(1)(1)	/访问第1个序列成员中的第1个成员：11
3	=A1(2)(2)	/访问第2个序列成员中的第2个成员：17

在SPL中输出之前示例中的点阵文字：E

	A	B	C	D
1	[[1,1,1],[1,0,0],[1,1,0],[1,0,0],[1,1,1]]			
2	for A1	=""		
3		for A2	if B3==1	>B2=B2+"@"
4			else	>B2=B2+" "
5		>output(B2)		

输出结果：

```
@@@
@
@@
@
@@@
```

8.5.3 多维序列的使用

【例8.18】用多维序列实现矩阵 $\begin{bmatrix} 8 & 3 \\ -4 & -5 \end{bmatrix}$ 与矩阵 $\begin{bmatrix} 9 & 7 \\ -1 & 2 \end{bmatrix}$ 的加法计算。

8.5.3 多维序列的使用

【例8.18】用多维序列实现矩阵 $\begin{bmatrix} 8 & 3 \\ -4 & -5 \end{bmatrix}$ 与矩阵 $\begin{bmatrix} 9 & 7 \\ -1 & 2 \end{bmatrix}$ 的加法计算。

解题思路

遍历A1中每个序列成员中的每个数值成员，再使用成员的序号找到B1中与之对应的数值成员，做加法计算后将结果保存于C1中对应的位置

	A	B	C	D
1	[[8,3],[-4,-5]]	[[9,7],[-1,2]]	[[0,0],[0,0]]	/A1：$\begin{bmatrix} 8 & 3 \\ -4 & -5 \end{bmatrix}$，B1：$\begin{bmatrix} 9 & 7 \\ -1 & 2 \end{bmatrix}$，C1保存结果
2	for A1	for A2	>C1(#A2)(#B2)=B2+B1(#A2)(#B2)	/用嵌套循环，按位置实现矩阵的加法

C1	Index	Member
	1	[17,10]
	2	[-5,-3]

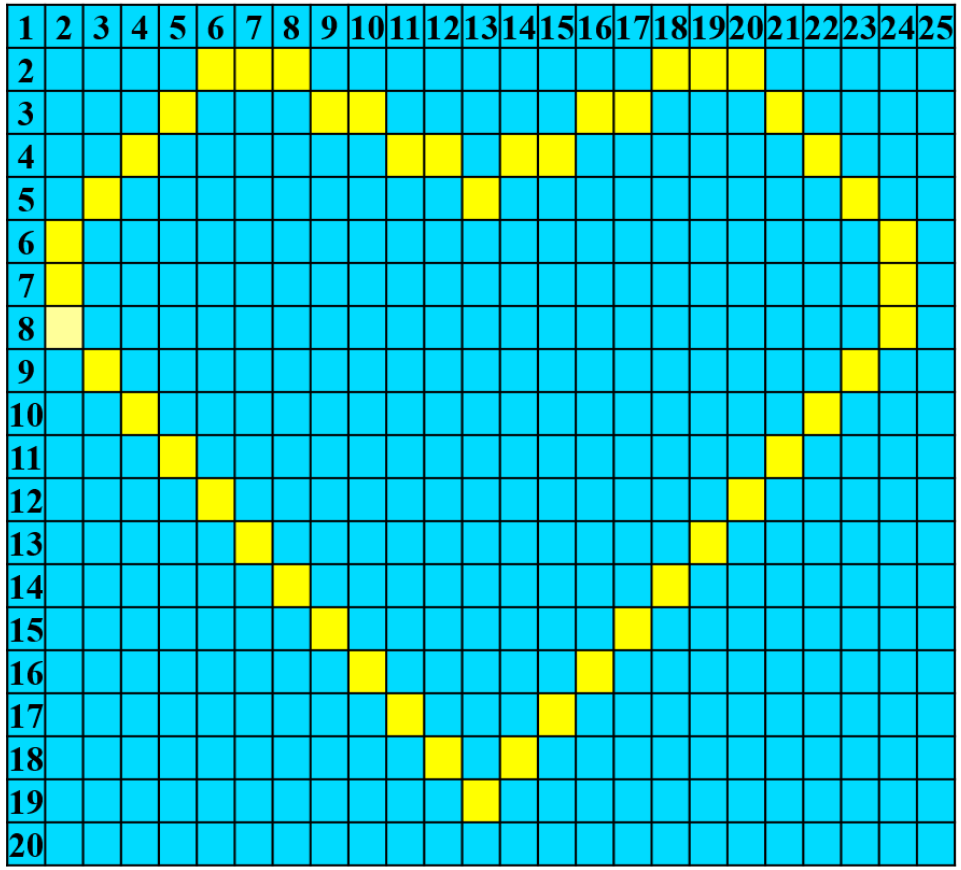
即计算结果：$\begin{bmatrix} 17 & 10 \\ -5 & -3 \end{bmatrix}$

8.5.3 多维序列的使用

【例8.19】小明想在情人节那天向在某大楼工作的小芳示爱。大楼管理人员答应在情人节当天将大楼的窗户通过亮灯与灭灯作成心形。大楼共20层每层有25个窗户，应如何记录亮灯与灭灯的房间?

8.5.3 多维序列的使用

【例8.19】小明想在情人节那天向在某大楼工作的小芳示爱。大楼管理人员答应在情人节当天将大楼的窗户通过亮灯与灭灯作成心形。大楼共20层每层有25个窗户，应如何记录亮灯与灭灯的房间?



解题思路

要处理这类问题，先画出直观的示意图，无疑是最方便的解决办法。

由示意图可知：这个问题与点阵显示文字，其实是同一类问题。所以同样可以使用多维序列来进行处理。

因为每行亮灯的房间较少，我们可以改进先前的方法：让多维序列中的序列成员只记录每行的亮灯房间所对应的数值。

8.5.3 多维序列的使用

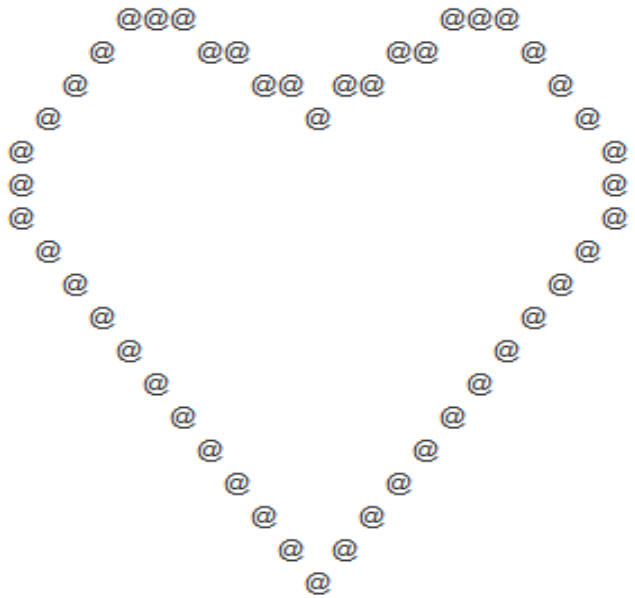
【例8.19】小明想在情人节那天向在某大楼工作的小芳示爱。大楼管理人员答应在情人节当天将大楼的窗户通过亮灯与灭灯作成心形。大楼共20层每层有25个窗户，应如何记述亮灯与灭灯的房间?

	A	B	C	D	E
1	[[],[6,7,8,18,19,20],[5,9,10,16,17,21],[4,11,12,14,15,22],[3,13,23],[2,24],[2,24],[2,24],[3,23],[4,22],[5,21],[6,20],[7,19],[8,18],[9,17],[10,16],[11,15],[12,14],[13],[]]	=[[0]*25,[0]*25]			/A1为记录每行亮灯的房间的多维矩阵，B1用来将其还原为包括所有房间的亮灭灯状态的点阵
2	for A1	for A2	>B1(#A2)(B2)=1		/还原所有房间的亮灭灯状态的点阵
3	for B1	=""			/使用示例中的方法检验显示的结果
4		for A3	if B4==1	>B3=B3+"@"	
5			else	>B3=B3+"　"	
6		>output(B3)			

8.5.3 多维序列的使用

【例8.19】小明想在情人节那天向在某大楼工作的小芳示爱。大楼管理人员答应在情人节当天将大楼的窗户通过亮灯与灭灯作成心形。大楼共20层每层有25个窗户，应如何记述亮灯与灭灯的房间?

SPL中的输出结果:



注意

- 实际使用中，应根据用于显示亮灯或灭灯的元素的显示尺寸不同，做出一些适当的调整。

8.6 习题

8.6 习题

【习题8.1】有员工工资记录（已缴纳社保：4500，6000，5500，8000，12000，8500，15000，7000，18000，25000，20000）与个税税率表（如图），假设已知公司最高工资2.5万元/月，求缴税后的各员工工资。

【习题8.2】有文件名为"filename@xxxx.com.rar"的字符串，想将"@xxxx.com"替换为"@yyyy.net"

【习题8.3】想坐火车从邯郸去沈阳，直达车票已售完需要中转一次。考虑到达沈阳的时间上的要求，共三个城市的部分车次可供中转选择：秦皇岛、唐山、保定。详细信息记录在序列中（A1中数据格式为：车次-中转站-到达时间，B1中数据格式为：车次-中转站-发车时间），想列出所有的可行方案（换乘车次，中转站）。

【习题8.4】小明和小红的物理成绩都比较突出，已知两人共四次的近期物理模拟考试成绩（小明：97，99，90，86；小红：93，94，91，96），四次考试根据难度系数而设定的权重依次为：0.23，0.30，0.21，0.26，据此分析应该派谁参加竞赛。

【习题8.5】某网站开发小组有三位成员，每人有不同分工：Dennis(Photoshop)，Rose(HTML、JavaScript)，Jimmy(PHP、MySQL)。现在Rose想离职，新招入的Mike将接手她的工作。想知道最新的工作安排情况。

【习题8.6】有两个序列，第一个保存人名（Mike，Tom，Andy），第二个保存地址（London，Manchester，Liverpool），两组数据一一对应，现在想要合并两个序列的内容并写到一个文件里。

【习题8.7】有一份匿名的已排好序的某学习小组的物理考试成绩（98，97，93，92，89，89，87，83，79，76，74，74，68，65，63，62，59，58，58，52），小刚想知道自己的成绩（79）的排名（假设没有其他人与小刚考一样的分数），怎样实现快速的查找？

【习题8.8】 已知：Dennis White，Rose Smith，John Davis，James Smith，Robert Jones，Jimmy Thomas。这几人中已知有且仅有两个姓氏相同的兄弟姐妹，试找出这两人。

【习题8.9】电影中，某特工要与线人接头，预先约定的接头地点有三处：东京（Tokyo），伦敦（London）或亚伦（Yaren）。特工某天收到了线人发来的电报，内容是5串数字：10001、01010、00100、00100、00100，这是线人在暗示他要去哪个城市与特工接头呢？

8.6 习题

【习题8.1】有员工工资记录（已缴纳社保：4500，6000，5500，8000，12000，8500，15000，7000，18000，25000，20000）与个税税率表（如图），假设已知公司最高工资2.5万元/月，求缴税后的各员工工资。

个税税率表：

1、低于5000元，无需缴纳个税
2、个税区间在0-3000元税率为3%
3、个税区间在3000元-12000元税率为10%
4、个税区间在12000元-25000元税率为20%
5、个税区间在25000元-35000元税率为25%
6、个税区间在35000元-55000元税率为30%
7、个税区间在55000元-80000元税率为35%
8、个税区间大于80000元税率为45%

8.6 习题

【习题8.1】有员工工资记录（已缴纳社保：4500，6000，5500，8000，12000，8500，15000，7000，18000，25000，20000）与个税税率表（如图），假设已知公司最高工资2.5万元/月，求缴税后的各员工工资。

个税税率表：

1、低于5000元，无需缴纳个税
2、个税区间在0-3000元税率为3%
3、个税区间在3000元-12000元税率为10%
4、个税区间在12000元-25000元税率为20%
5、个税区间在25000元-35000元税率为25%
6、个税区间在35000元-55000元税率为30%
7、个税区间在55000元-80000元税率为35%
8、个税区间大于80000元税率为45%

	A	B	C	D
1	[4500,6000,5500,8000,12000,8500,15000,7000,18000,25000,20000]	=(3000-0)*0.03	=(12000-3000)*0.1	/工资记录
2	for A1	if A2>5000 && A2<=8000	>A1(#A2)=A2-(A2-5000)*0.03	/税率第2档
3		else if A2>8000 && A2<=17000	>A1(#A2)=A2-B1-(A2-8000)*0.10	/税率第3档
4		else if A2>17000 && A2<=30000	>A1(#A2)=A2-B1-C1-(A2-17000)*0.20	/税率第4档

A1	Index	Member
	1	4500
	2	5970.0
	3	5485.0
	4	7910.0
	5	11510.0
	6	8360.0
	7	14210.0
	8	6940.0
	9	16810.0
	10	22410.0
	11	18410.0

8.6 习题

【习题8.2】有文件名为"filename@xxxx.com.rar"的字符串，想将"@xxxx.com"替换为"@yyyy.net"

8.6 习题

【习题8.2】有文件名为"filename@xxxx.com.rar"的字符串，想将"@xxxx.com"替换为"@yyyy.net"

	A	B	C
1	filename@xxxx.com.rar		/原始字符串
2	=[mid(A1,1,pos@z(A1,"@")-1), "@yyyy.net", mid(A1,pos@z(A1,"."))]		/剪出需要的子串，存入序列
3	for A2	>B1=B1/A3	/遍历字符串序列，重新拼接

B1	Value
	filename@yyyy.net.rar

8.6 习题

【习题8.3】想坐火车从邯郸去沈阳，直达车票已售完需要中转一次。考虑到达沈阳的时间上的要求，共三个城市的部分车次可供中转选择：秦皇岛、唐山、保定。详细信息记录在序列中（A1中数据格式为：车次-中转站-到达时间，B1中数据格式为：车次-中转站-发车时间），想列出所有的可行方案（换乘车次，中转站）。

8.6 习题

【习题8.3】想坐火车从邯郸去沈阳，直达车票已售完需要中转一次。考虑到达沈阳的时间上的要求，共三个城市的部分车次可供中转选择：秦皇岛、唐山、保定。详细信息记录在序列中（A1中数据格式为：车次-中转站-到达时间，B1中数据格式为：车次-中转站-发车时间），想列出所有的可行方案（换乘车次，中转站）。

	A	B	C	D	E
1	[[G6272,Baoding,11:28:00],[G6272,Qinhuangdao,14:42:00],[G6272,Tangshan,13:46:00]]	[[G1290,Baoding,14:16:00],[G1221,Qinhuangdao,15:49:00],[G1221,Tangshan,15:01:00]]	[]		/中转信息
2	for A1	for B1	if A2(2) == B2(2) && time(A2(3)) < time(B2(3))	>C1=C1|(A2(1)/"->"/B2(1)/", Station : "/A2(2))	/匹配合适的换乘车次与换乘地点并存储结果

C1	Index	Member
	1	G6272->G1290, Station : Baoding
	2	G6272->G1221, Station : Qinhuangdao
	3	G6272->G1221, Station : Tangshan

8.6 习题

【习题8.4】小明和小红的物理成绩都比较突出，已知两人共四次的近期物理模拟考试成绩（小明：97，99，90，86；小红：93，94，91，96），四次考试根据难度系数而设定的权重依次为：0.23，0.30，0.21，0.26，据此分析应该派谁参加竞赛。

8.6 习题

【习题8.4】小明和小红的物理成绩都比较突出，已知两人共四次的近期物理模拟考试成绩（小明：97，99，90，86；小红：93，94，91，96），四次考试根据难度系数而设定的权重依次为：0.23，0.30，0.21，0.26，据此分析应该派谁参加竞赛。

	A	B	C	D	E
1	[97,99,90,86]	[93,94,91,96]	[0.23,0.30,0.21,0.26]	0	0
2	for A1	>D1=D1+A2*C1(#A2)	/D1为小明的得分评估		
3	for B1	>E1=E1+A3*C1(#A3)	/E1为小红的得分评估		

小明：

D1	Value
	93.27

小红：

E1	Value
	93.66

从最终结果来看小红的优势更大，应该派她去参加竞赛。

8.6 习题

【习题8.5】某网站开发小组有三位成员，每人有不同分工：Dennis(Photoshop)，Rose(HTML、JavaScript)，Jimmy(PHP、MySQL)。现在Rose想离职，新招入的Mike将接手她的工作。想知道最新的工作安排情况。

8.6 习题

【习题8.5】某网站开发小组有三位成员，每人有不同分工：Dennis(Photoshop)，Rose(HTML、JavaScript)，Jimmy(PHP、MySQL)。现在Rose想离职，新招入的Mike将接手她的工作。想知道最新的工作安排情况。

	A	B	C	D
1	[[Dennis,Photoshop], [Rose, HTML], [Jimmy, PHP],[Jimmy,MySQL], [Rose,JavaScript]]			/小组成员工作分配信息
2	for A1	if A2(1) == "Rose"	>A1(#A2)(1) = "Mike"	/将名单中的Rose全部替换成Mike

A1	Index	Member
	1	[Dennis,Photoshop]
	2	[Mike,HTML]
	3	[Jimmy,PHP]
	4	[Jimmy,MySQL]
	5	[Mike,JavaScript]

8.6 习题

【习题8.6】有两个序列，第一个保存人名（Mike，Tom，Andy），第二个保存地址（London，Manchester，Liverpool），两组数据一一对应，现在想要合并两个序列的内容并写到一个文件里。

8.6 习题

【习题8.6】有两个序列，第一个保存人名（Mike，Tom，Andy），第二个保存地址（London，Manchester，Liverpool），两组数据一一对应，现在想要合并两个序列的内容并写到一个文件里。

	A	B	C
1	[Mike, Tom, Andy]	[London, Manchester, Liverpool]	/定义数列A1和B1
2	for A1	>A1(#A2)=A2/":"/B1(#A2)	/将字符串依次对位进行拼接
3	>file("D:/result.txt").export(A1)		/将A1写入文件，每行一个

结果：

```
Mike:London
Tom:Manchester
Andy:Liverpool
```

8.6 习题

【习题8.7】有一份匿名的已排好序的某学习小组的物理考试成绩（98，97，93，92，89，89，87，83，79，76，74，74，68，65，63，62，59，58，58，52），小刚想知道自己的成绩（79）的排名（假设没有其他人与小刚考一样的分数），怎样实现快速的查找?

8.6 习题

【习题8.7】有一份匿名的已排好序的某学习小组的物理考试成绩（98，97，93，92，89，89，87，83，79，76，74，74，68，65，63，62，59，58，58，52），小刚想知道自己的成绩（79）的排名（假设没有其他人与小刚考一样的分数），怎样实现快速的查找?

	A	B	C	D	E
1	[98, 97, 93, 92, 89, 89, 87, 83, 79, 76, 74, 74, 68, 65, 63, 62, 59, 58, 58, 52]	79	1	=A1.len()	/A1是全体考试成绩，B1是小刚考试成绩，C1是查询范围上限，D1是查询范围下限
2	for C1<=D1	=int((C1+D1)/2)	=A1(B2)		/计算查询上下限的中位值并取整
3		if C2==B1	>output(B2)	break	/如果中位值等于待查数值则输出
4		else if C2<B1	>D1=B2-1		/如果中位值小于待查数值则更新范围下限
5		else	>C1=B2+1		/如果中位值小于待查数值则更新范围上限

结果：

9

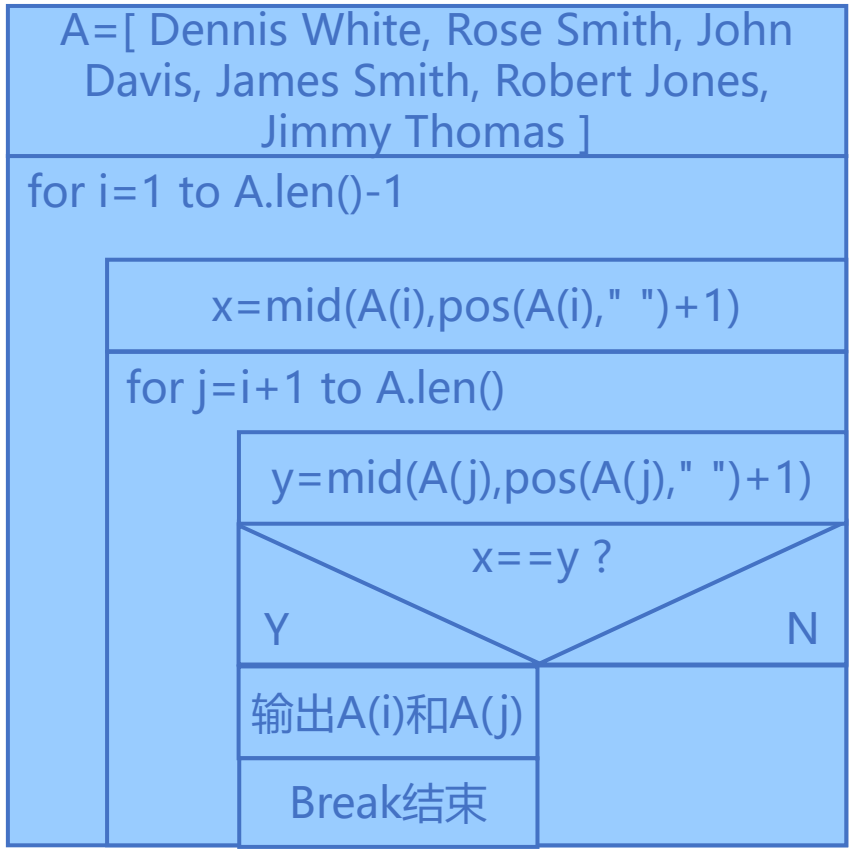
即小刚的成绩排在小组的第9名。
若代码无输出则说明分数不在列表中。

8.6 习题

【习题8.8】 已知：Dennis White，Rose Smith，John Davis，James Smith，Robert Jones，Jimmy Thomas。这几人中已知有且仅有两个姓氏相同的兄弟姐妹，试找出这两人。

8.6 习题

【习题8.8】 已知：Dennis White，Rose Smith，John Davis，James Smith，Robert Jones，Jimmy Thomas。这几人中已知有且仅有两个姓氏相同的兄弟姐妹，试找出这两人。



	A	B	C	D	E
1	[Dennis White, Rose Smith, John Davis, James Smith, Robert Jones, Jimmy Thomas]				
2	for A1.len()-1	=mid(A1(A2),pos(A1(A2)," ")+1)			/依次取出姓氏
3		for A2+1, A1.len(	=mid(A1(B3),pos(A1(B3)," ")+1)		/向后查找比较
4			if B2==C3	>B1=(A1(A2)/" and "/A1(B3))	/如有姓氏相同输出两个人名
5				break A2	/结束所有循环

B1	Value
	Rose Smith and James Smith

8.6 习题

【习题8.9】电影中，某特工要与线人接头，预先约定的接头地点有三处：东京（Tokyo），伦敦（London）或亚伦（Yaren）。特工某天收到了线人发来的电报，内容是5串数字：10001、01010、00100、00100、00100，这是线人在暗示他要去哪个城市与特工接头呢?

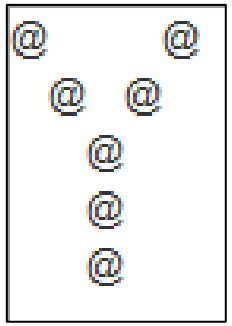
8.6 习题

【习题8.9】电影中，某特工要与线人接头，预先约定的接头地点有三处：东京（Tokyo），伦敦（London）或亚伦（Yaren）。特工某天收到了线人发来的电报，内容是5串数字：10001、01010、00100、00100、00100，这是线人在暗示他要去哪个城市与特工接头呢?

解题思路

这5串可疑的数字显然很像点阵，我们可以使用第四章中例题的办法将其显示出来查看结果。

输出结果：



显示字母"Y"，即接头地点在亚伦（Yaren）

	A	B	C	D
1	[[1,0,0,0,1],[0,1,0,1,0],[0,0,1,0,0],[0,0,1,0,0],[0,0,1,0,0]]			
2	for A1	=""		
3		for A2	if B3==1	>B2=B2+"@"
4			else	>B2=B2+" "
5		>output(B2)		

第9章 序列的计算

9.1 序列的集合性

- 序列的有序性
- 序列与集合

9.2 序列函数

- 序列的生成与维护
- 字符串与序列

9.3 聚合计算

- 计数、求和与均值
- 排名
- 极值和中位数
- 其它聚合函数

9.4 循环计算

- 用A.(x)完成循环计算
- 定位计算
- 循环函数

9.5 定位

- 查找指定成员的位置
- 查找满足条件成员的位置
- 查找最大/最小值的位置
- 整体定位

9.6 选出

- 查找指定成员的位置
- 最大/最小值选出
- 排序

9.7 多维序列的计算

9.8 序列计算举例

9.9 习题

9.1 序列的集合性

9.1.1 序列的有序性

有序/无序集合差别

- 一般的集合是**无序**的，即：成员相同但顺序不同的两个集合是相等的。但**序列**具有**有序性**，成员相同但**顺序不同**的两个序列**不相等**。此外，序列中的成员是**可重复**的。
- 因为序列是有序的，所以可以用**cmp(A, B)**函数进行**对位比较大小的运算**，以及使用形如**A??B**的**对位四则计算**。
- 如果只需比较两个序列内的成员是否一致，而不考虑顺序，可以用A.eq(B)判断。

有序性

	A	B
1	=[1,2]==[2,1]	/false
2	=[1,2].eq([2,1])	/true
3	=[1,2]==[1,2]	/true

	A	B
1	=[1]==[1,1]	/false
2	=[1,1]==[1,1]	/true

9.1.1 序列的有序性

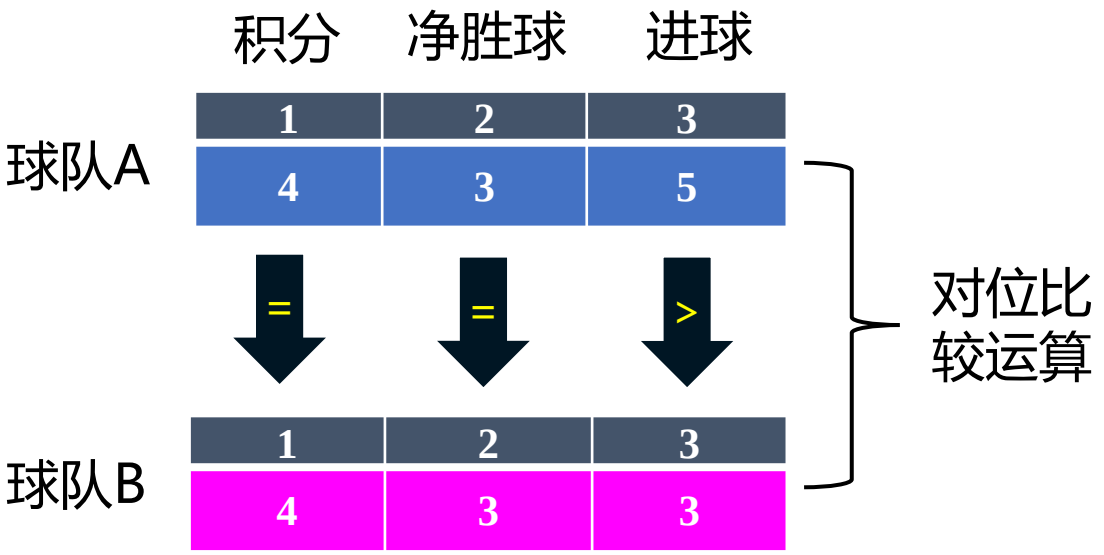
【例9.1】世界杯小组赛中，决定名次的首要条件是各队在全部三场小组比赛中的积分。积分相同的情况下，净胜球（总进球减去总失球）较多的球队排名靠前。如果净胜球仍然相同，总进球较多的球队排名靠前。现在用序列按顺序记录各球队的积分、净胜球数和进球数（A队：积分4，净胜球数3，进球数5；B队：积分4，净胜球数3，进球数3），如何快速判定两个球队在小组内的先后排名？

9.1.1 序列的有序性

【例9.1】世界杯小组赛中，决定名次的首要条件是各队在全部三场小组比赛中的积分。积分相同的情况下，净胜球（总进球减去总失球）较多的球队排名靠前。如果净胜球仍然相同，总进球较多的球队排名靠前。现在用序列按顺序记录各球队的积分、净胜球数和进球数（A队：积分4，净胜球数3，进球数5；B队：积分4，净胜球数3，进球数3），如何快速判定两个球队在小组内的先后排名？

解题思路

可以将积分，净胜球数和进球数按顺序存入一个序列，然后使用**cmp(A, B)**函数将两个序列对位比较。



	A
1	=cmp([4,3,5],[4,3,3])

结果：

A1	Value
	1

即：球队A排名高于球队B。

注意

- 序列比较时，将依次对位比较每个成员的值，遇到第一个不等成员时根据大小分别返回1或-1，全相等则返回0。
- 比较时对应成员的数据类型必须是可以进行比较的。

9.1.1 序列的有序性

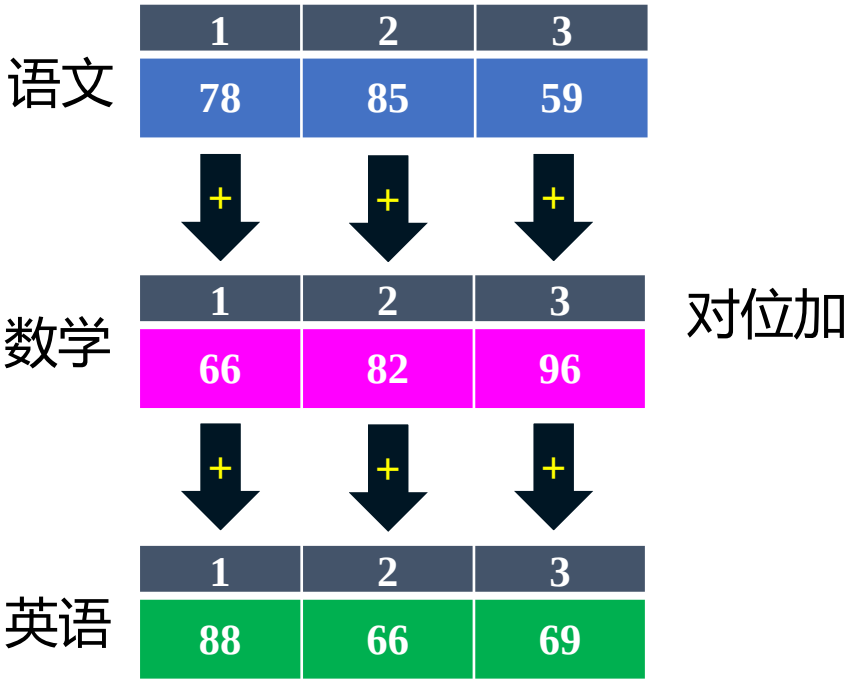
【例9.2】已知3名学生某次测验的分数依次如下：语文：78，85，59；数学：66，82，96；英语：88，66，69。求每个人的总成绩。

9.1.1 序列的有序性

【例9.2】已知3名学生某次测验的分数依次如下：语文：78，85，59；数学：66，82，96；英语：88，66，69。求每个人的总成绩。

解题思路

可以将3人各科成绩分别保存于序列中，再使用对位加（++）计算，分别累加后算出每个人的总成绩。



	A	B	C
1	[78,85,59]	[66,82,96]	[88,66,69]
2	=A1++B1++C1		

A2	Index	Member
	1	232
	2	233
	3	224

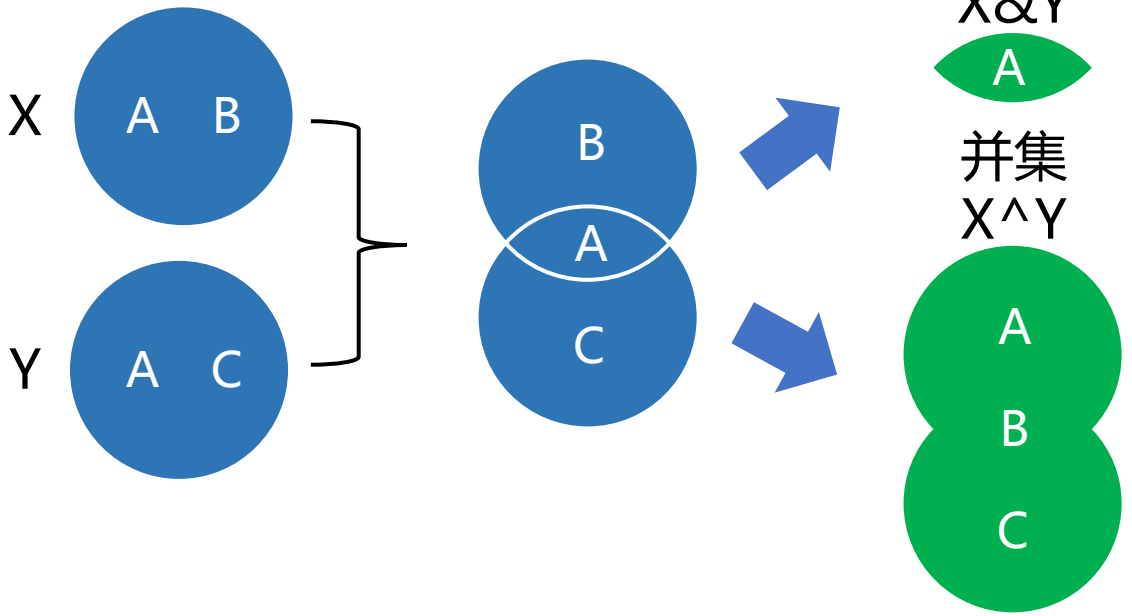
注意

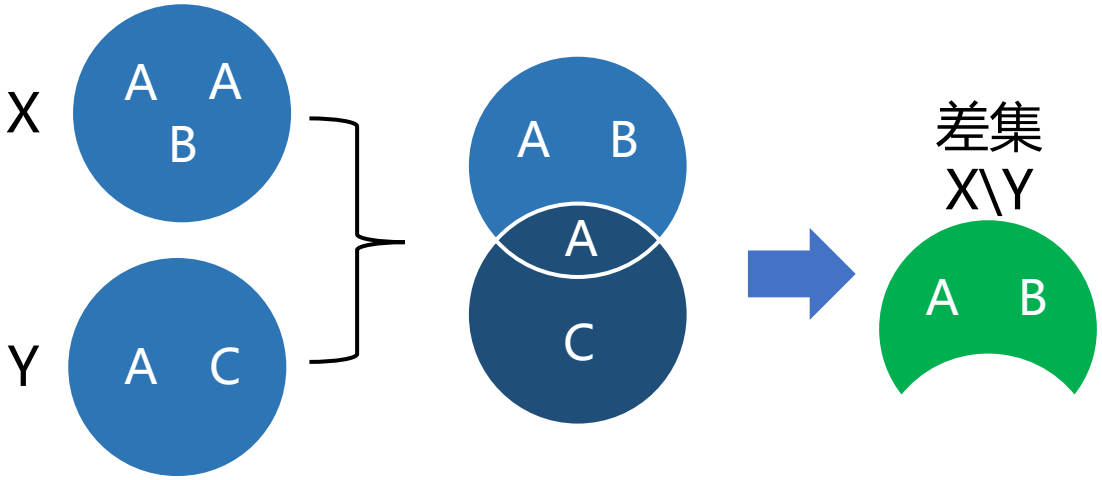
- 序列的对位四则运算，即**长度相同**的两个序列，依次将相同位置的成员进行**对位计算**，返回新序列。
- 对位四则计算包括：对位加++、对位减--、对位乘**、对位除//，对位整除\\以及对位求余%%

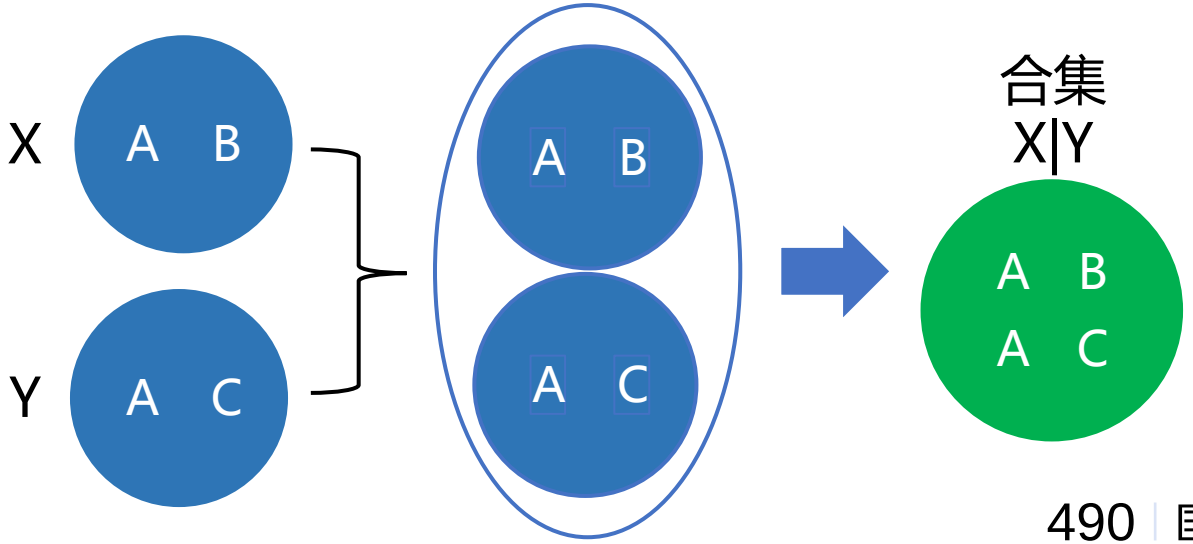
9.1.2 序列与集合

集合运算

- 序列具有**集合**一般特性，可以进行集合运算。
- 集合运算包括：**交集(&)、并集(^)、差集(\)、合集(|)**等。







交集

- 左侧序列中的成员若在右侧序列中出现，则**依序**加入新序列中（**取同**），否则略过不计。
- 右侧序列中的每个成员，都只能用于**一次**取同运算，不可以多次使用。

	A
1	=[4,4,3] ^ [1,3,4]
2	=[4,3] ^ [1,3,4,4]
3	=[4,4,3] ^ [1,3,4,4]

A1	Index	Member
	1	4
	2	3

A2	Index	Member
	1	4
	2	3

A3	Index	Member
	1	4
	2	4
	3	3

【例9.3】某班报名课外活动兴趣班，有写作与绘画2种选择（可同时报2个）。已知报名写作的有：Jimmy、Mary、Lucy，报名绘画的有Kate、Tom、Mary，问报了2个兴趣班的都有谁?

【例9.3】某班报名课外活动兴趣班，有写作与绘画2种选择（可同时报2个）。已知报名写作的有：Jimmy、Mary、Lucy，报名绘画的有Kate、Tom、Mary，问报了2个兴趣班的都有谁?

解题思路

将每个兴趣班的报名名单创建为1个序列，再求出2个序列的**交集**即可。

	A	B	C
1	[Jimmy, Mary, Lucy]	[Kate, Tom, Mary]	/A1：写作报名，B1：绘画报名
2	=A1^B1		/求A1与B1的交集

A2	Index	Member
	1	Mary

并集

- 先将左侧序列中的成员**依序**加入新序列中，右侧序列中的成员若在左侧序列中出现，则略过不计（**去重**），否则**依序**加入新序列中。
- 左侧序列中的每个成员，都只能用于**一次**去重运算，不可以多次使用。

	A
1	=[4,3] & [3,4]
2	=[4,4,3] & [3,4]
3	=[4,3] & [3,4,4]

A1	Index	Member
	1	4
	2	3

A2	Index	Member
	1	4
	2	4
	3	3

A3	Index	Member
	1	4
	2	3
	3	4

【例9.4】某班报名课外活动兴趣班，有写作与绘画2种选择（可同时报2个）。已知报名写作的有：Jimmy、Mary、Lucy，报名绘画的有Kate、Tom、Mary，报名参加兴趣班的总共有几人?

并集

【例9.4】某班报名课外活动兴趣班，有写作与绘画2种选择（可同时报2个）。已知报名写作的有：Jimmy、Mary、Lucy，报名绘画的有Kate、Tom、Mary，报名参加兴趣班的总共有几人?

解题思路

将每个兴趣班的报名名单创建为1个序列，求出2个序列的**并集**，再求出并集的**序列长度**即可。

	A	B	C
1	[Jimmy, Mary, Lucy]	[Kate, Tom, Mary]	/A1：写作报名，B1：绘画报名
2	=A1&B1		/求A1与B1的并集
3	=A2.len()		/求A2的序列长度

A3	Value
	5

差集

- 左侧序列中的成员若在右侧序列中出现，则略过不计，否则**依序**加入新序列中（**取异**）。
- 右侧序列中的每个成员，都只能用于**一次**取异运算，不可以多次使用。

	A
1	=[4,4,3] \ [1,4]
2	=[4,4,3] \ [1,3,4]
3	=[1,3,4] \ [4,4,3]

A1	Index	Member
	1	4
	2	3

A2	Index	Member
	1	4

A3	Index	Member
	1	1

【例9.5】某班报名课外活动兴趣班，有写作与绘画2种选择（可同时报2个）。已知报名写作的有：Jimmy、Mary、Lucy，报名绘画的有Kate、Tom、Mary， 只报名1个兴趣班的总共有几人?

差集

【例9.5】某班报名课外活动兴趣班，有写作与绘画2种选择（可同时报2个）。已知报名写作的有：Jimmy、Mary、Lucy，报名绘画的有Kate、Tom、Mary， 只报名1个兴趣班的总共有几人?

解题思路

将每个兴趣班的报名名单创建为1个序列，先求出2个序列的交集，即同时报了2个兴趣班的。再求出2个序列的并集，即所有报了兴趣班的。然后求出并集与交集的**差集**，即只报了1个兴趣班的。最后求出差集的**序列长度**即可。

	A	B	C
1	[Jimmy, Mary, Lucy]	[Kate, Tom, Mary]	/A1：写作报名，B1：绘画报名
2	=A1^B1	=A1&B1	/求A1与B1的交集与并集
3	=B2\A2		/求B2与A2的差集
4	=A3.len()		/求A3的序列长度

A4	Value
	4

合集

- 先将左侧序列中的成员**依序**加入新序列中，再将右侧序列中的成员**依序**加入新序列中。相同成员可以**重复**出现。
- 也可以使用非序列类型的“单值”数据，运算时将自动转换为含有**单一成员**的**序列**。

	A
1	=[4,3] | [1,4]
2	=[4,3] | 4
3	=4 | [4,3]

A1	Index	Member
	1	4
	2	3
	3	1
	4	4

A2	Index	Member
	1	4
	2	3
	3	4

A3	Index	Member
	1	4
	2	4
	3	3

【例9.6】小明和小红去吃西餐，小明点了1个牛排(Steak)、1个沙拉(Salad)、1个披萨(Pizza)，小红点了1个炸鸡腿(Drumsticks)、1个罗宋汤(Borsch)、1个牛排(Steak)，问他们一共点了哪些菜?

9.1.2 序列与集合

合集

【例9.6】小明和小红去吃西餐，小明点了1个牛排(Steak)、1个沙拉(Salad)、1个披萨(Pizza)，小红点了1个炸鸡腿(Drumsticks)、1个罗宋汤(Borsch)、1个牛排(Steak)，问他们一共点了哪些菜?

解题思路

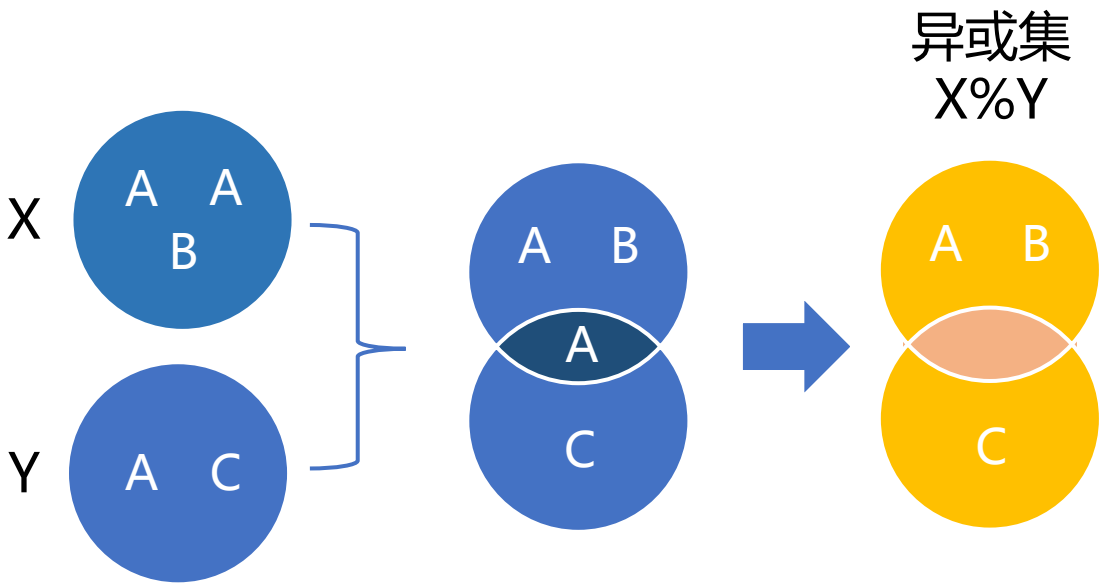
将小明和小红点的菜单分别创建为1个序列，显然重复点的菜应该分开计算，所以应该求两个序列的**合集**。

	A	B	C
1	[Steak, Salad, Pizza]	[Drumsticks, Borsch, Steak]	/A1：小明点的菜，B1：小红点的菜
2	=A1|B1		/求A1与B1的合集

A2	Index	Member
	1	Steak
	2	Salad
	3	Pizza
	4	Drumsticks
	5	Borsch
	6	Steak

异或集

- **属于**两个集合，但**不同时属于**两个集合的成员，它们构成的集合称为**异或集。**
- 两个序列在计算异或列时，左侧序列中**未出现**在右侧序列中的成员，**依序**加入新序列中；然后再将右侧序列中**未出现**在左侧序列中的成员**依次**加入。



【例9.5】某班报名课外活动兴趣班，有写作与绘画2种选择（可同时报2个）。已知报名写作的有：Jimmy、Mary、Lucy，报名绘画的有Kate、Tom、Mary， 只报名1个兴趣班的总共有几人?

【例9.5】某班报名课外活动兴趣班，有写作与绘画2种选择（可同时报2个）。已知报名写作的有：Jimmy、Mary、Lucy，报名绘画的有Kate、Tom、Mary， 只报名1个兴趣班的总共有几人?

解题思路

用集合的异或运算，例9.5可以直接计算。

	A	B	C
1	[Jimmy, Mary, Lucy]	[Kate, Tom, Mary]	/A1：写作报名，B1：绘画报名
2	=A1%B1		/求A1与B1的异或集
3	=A2.len()		/求A2的序列长度

A2	Index	Member
	1	Jimmy
	2	Lucy
	3	Kate
	4	Tom

A3	Value
	4

9.2 序列的生成与维护

9.2.1 序列的生成与维护

序列的生成

SPL中，序列的生成常用方法有**常数构造法**和**函数构造法**。

	A	B
1	[2,3,4]	/常数构造法生成2~4的数列，也可以用在表达式中，写作=[2,3,4]
2	=to(2,4)	/使用to(a,b)函数生成a到b的数列。如果省略一个参数，也可以用to(n)生成1到n的数列。
3	=["a", "b", "c", "d"].to(2,4)	/使用A.to()函数从已有序列中正向取第2个到第4个成员构成新序列；字符串序列在表达式中构造时，成员需加引号
4	=[1,2,3,4,5].step(3,1,2)	/使用A.step()函数在已有序列中，取出每3个成员中的第1个和第2个构成新序列

A1	Index	Member
	1	2
	2	3
	3	4

A2	Index	Member
	1	2
	2	3
	3	4

A3	Index	Member
	1	b
	2	c
	3	d

A4	Index	Member
	1	1
	2	2
	3	4
	4	5

9.2.1 序列的生成与维护

【例9.7】某部门员工名单如下：David, Daniel, Andrew, Ruby, Rachel, Rebecca。准备派遣前3名员工外出学习，请用序列展示派遣员工的名单。

9.2.1 序列的生成与维护

【例9.7】某部门员工名单如下：David, Daniel, Andrew, Ruby, Rachel, Rebecca。准备派遣前3名员工外出学习，请用序列展示派遣员工的名单。

解题思路

员工名单可以用序列记录，派遣员工名单只需获取前3项构成新序列即可。

A.to(n)可以从已有序列**A**中，截取前**n**个成员生成新序列。

	A	B
1	[David, Daniel, Andrew, Ruby, Rachel, Rebecca]	/用常数构造法生成员工列表
2	=A1.to(3)	/获取前3位员工得到派遣名单

A2	Index	Member
	1	David
	2	Daniel
	3	Andrew

9.2.1 序列的生成与维护

【例9.8】某部门员工名单如下：David, Daniel, Andrew, Ruby, Rachel, Rebecca。先选出第1位员工，之后每隔1人选出1名员工外出学习，请列出派遣员工的名单。

9.2.1 序列的生成与维护

【例9.8】某部门员工名单如下：David, Daniel, Andrew, Ruby, Rachel, Rebecca。先选出第1位员工，之后每隔1人选出1名员工外出学习，请列出派遣员工的名单。

解题思路

将初始员工序列每隔1个成员取1个成员，生成新的序列即可。

A.step(n, k_1, k_2, ...)可以按照指定的起始序号位置**k_1, k_2,...**和跨度**n**，生成新的序列。**k_1, k_2,...**省略时起始序号为1。

	A	B
1	[David, Daniel, Andrew, Ruby, Rachel, Rebecca]	/用常数构造法生成员工列表
2	=A1.step(2)	/派遣名单每隔1个成员取1个成员

A2	Index	Member
	1	David
	2	Andrew
	3	Rachel

9.2.1 序列的生成与维护

序列的维护

SPL中，序列的维护包括序列中成员的**插入、修改**与**删除**，分别使用**A.insert(), A.modify()**和**A.delete()**函数。

	A	B
1	=[1,2,3].insert(0,4)	/末尾插入4
2	=[1,4].insert(2,[2,3])	/第1~2位成员之间依次插入2和3
3	=[1,2,3].modify(2,0)	/修改第2个成员
4	=[1,2,3].modify(2,[-2,-3])	/从第2位开始修改
5	=[1,2,3].delete(1)	/删除第1位
6	=[1,2,3].delete([1,3])	/删除第1位和第3位

A1	Index	Member
	1	1
	2	2
	3	3
	4	4

A2	Index	Member
	1	1
	2	2
	3	3
	4	4

A3	Index	Member
	1	1
	2	0
	3	3

A4	Index	Member
	1	1
	2	-2
	3	-3

A5	Index	Member
	1	2
	2	3

A6	Index	Member
	1	2

9.2.1 序列的生成与维护

【例9.9】某部门员工名单如下：David, Daniel, Andrew, Ruby, Rachel, Rebecca。现在新招入1名员工Ashley，请列出现在员工的名单。

9.2.1 序列的生成与维护

【例9.9】某部门员工名单如下：David, Daniel, Andrew, Ruby, Rachel, Rebecca。现在新招入1名员工Ashley，请列出现在员工的名单。

解题思路

在初始员工序列中追加一个新成员即可。

A.insert(k, x)可以在序列**A**的第**k**个位置起插入成员**x**，**x**可以是单值或者序列；**k**为0时在序列最后添加。

	A	B
1	[David, Daniel, Andrew, Ruby, Rachel, Rebecca]	/用常数构造法生成初始员工名单
2	=A1.insert(0, "Ashley")	/名单中追加一个新成员

A2	Index	Member
	1	David
	2	Daniel
	3	Andrew
	4	Ruby
	5	Rachel
	6	Rebecca
	7	Ashley

注意

- **A.insert(k, x)**会将新的成员**x**插入到序列**A**中的第**k**个位置，序列中第**k**个位置后面的成员向后顺移。插入后**A会被修改**，所以上面的代码执行后A1和A2中结果相同。
- 在序列最后添加成员时，也可以用**|**操作符，如将A2中代码改为**=A1=A1|"Ashley"**，结果是相同的。

9.2.1 序列的生成与维护

【例9.10】某部门员工名单如下：David, Daniel, Andrew, Ruby, Rachel, Rebecca。现在发现其中第4位员工的名字登记错误，实际应该是Rudy，请列出修改后的员工名单。

9.2.1 序列的生成与维护

【例9.10】某部门员工名单如下：David, Daniel, Andrew, Ruby, Rachel, Rebecca。现在发现其中第4位员工的名字登记错误，实际应该是Rudy，请列出修改后的员工名单。

解题思路

在员工名单中修改一个指定位置的成员即可。

A.modify(k, x)可以从指定的位置**k**起，给序列**A**中一个或多个成员赋值，**x**可以是单值或者序列。

	A	B
1	[David, Daniel, Andrew, Ruby, Rachel, Rebecca]	/用常数构造法生成初始员工名单
2	=A1.modify(4, "Rudy")	/最终名单中修改一个指定成员的值

A2	Index	Member
	1	David
	2	Daniel
	3	Andrew
	4	Rudy
	5	Rachel
	6	Rebecca

注意

- **A.modify(k, x)**执行后**A会被修改**，所以上面的代码执行后A1和A2中结果相同。

9.2.1 序列的生成与维护

【例9.11】某部门员工名单如下：David, Daniel, Andrew, Ruby, Rachel, Rebecca。现在其中第3位和第5位员工需要转入其它部门，请列出人员变动后的名单。

9.2.1 序列的生成与维护

【例9.11】某部门员工名单如下：David, Daniel, Andrew, Ruby, Rachel, Rebecca。现在其中第3位和第5位员工需要转入其它部门，请列出人员变动后的名单。

解题思路

在员工清单中删除指定位置的成员即可。

A.delete(p)可以删除序列**A**中序号在数列**p**中的成员，只需删除一个成员时**p**也可以为整数。

	A	B
1	[David, Daniel, Andrew, Ruby, Rachel, Rebecca]	/用常数构造法生成初始员工名单
2	>A1.delete([3, 5])	/最终名单中删除两个成员的值

A1	Index	Member
	1	David
	2	Daniel
	3	Ruby
	4	Rebecca

注意

- **A.delete(p)**删除序列中的成员后，其它成员的位置会向前顺移补位。删除后**A会被修改**，所以上面的代码执行后A1和A2中结果相同。

9.2.2 字符串与序列

字符串与序列的相互转化

- s.split(d)可以将字符串s通过分隔符d拆成序列，无分隔符时每个字符拆分为一个成员。
- A.concat(d)可以用分隔符d将序列A中成员连接成字符串返回，无分隔符则直接拼接成字符串。

	A	B
1	="abc".split()	/按字符拆分成序列
2	=“a,b,c".split(",")	/以逗号作分隔符拆分成序列
3	="1,1.5,dc,null".split@p(",")	/以逗号作分隔符拆分并自动识别数据类型

A1	Index	Member
	1	a
	2	b
	3	c

A2	Index	Member
	1	a
	2	b
	3	c

A3	Index	Member
	1	1
	2	1.5
	3	dc
	4	(null)

	A	B
1	=["a","b","c","d","e"].concat()	/序列中每个成员依次连成一个字符串
2	=["a","b","c","d","e"].concat(",")	/序列成员之间以逗号连接

A1	Value
	abcde

A2	Value
	a,b,c,d,e

【例9.12】假设想考单词antidisestablishmentarianism的默写填空，需要把它的元音字母都替换为”_”号，再以字符串返回。

9.2.2 字符串与序列

【例9.12】假设想考单词antidisestablishmentarianism的默写填空，需要把它的元音字母都替换为”_”号，再以字符串返回。

解题思路

用**s.split()**函数将单词分割为字符串序列，将元音字母都替换后，再拼接返回。

	A	B	C	D
1	="antidisestablishmentarianism".split()			/拆分单词，构成字符序列
2	for A1	if A2=="a" || A2=="e" || A2=="i" || A2=="o" || A2=="u"	>A1(#A2) = "_"	/替换元音字符为"_”,辅音字符不变
3	=A1.concat()			/拼接字符串序列返回为一个字符串

A3	Value
	_nt_d_s_st_bl_shm_nt_r__n_sm

9.2.2 字符串与序列

【例9.13】输入一句英文，依次输出句子中的每个英文单词。如输入： The quick brown fox jumps over a lazy dog.

9.2.2 字符串与序列

【例9.13】输入一句英文，依次输出句子中的每个英文单词。如输入： The quick brown fox jumps over a lazy dog.

解题思路

用**s.split(d)**函数，用空格将句子分割为单词序列，但要注意需要先把句点等符号去除。

	A	B
1	The quick brown fox jumps over a lazy dog.	/输入句子
2	=left(A1, len(A1)-1)	/去除最后的句点
3	=A2.split(" ")	/用空格作为分隔符拆分出单词序列

A3	Index	Member
	1	The
	2	quick
	3	brown
	4	fox
	5	jumps
	6	over
	7	a
	8	lazy
	9	dog

9.2.2 字符串与序列

【例9.13】输入一句英文，依次输出句子中的每个英文单词。如输入： The quick brown fox jumps over a lazy dog.

注意

- 将英文句子中的单词拆分出来，可以使用更简单的函数**s.words()**，此时会自动拆分出单词序列，不必再另行处理句点等符号的影响了，同时单词中间的分隔符可以是空格，也可以是其它标点符号。

	A	B
1	The quick brown fox jumps over a lazy dog.	/输入句子
2	=A1.words()	/拆分出句子中的单词

A2	Index	Member
	1	The
	2	quick
	3	brown
	4	fox
	5	jumps
	6	over
	7	a
	8	lazy
	9	dog

9.3 聚合计算

9.3 聚合计算

聚合计算

- 针对集合中的所有成员统计出聚合结果，这样的运算称为**聚合运算**。
- 常用的聚合计算包括**计数、求和、求平均值、求极值、求排名**等。
- 聚合运算可以通过**循环结构**的代码来实现，但更方便的做法是使用**聚合函数**。

Index	Member
1	1
2	4
3	9
4	16
5	25
6	36

求和

	A	B	C
1	[1,4,9,16,25,36]	0	/A1输入序列，B1记录和
2	for A1	>B1=B1+A2	/循环求和

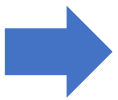
B1	Value
	91

9.3.1 计数、求和与均值

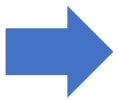
计数

- **计数**是指统计序列中成员的**总个数**，使用聚合函数**A.count()**计算。
- 与序列长度函数**A.len()不同**的是，计数时**不计序列中的空成员**。
- 如果序列中包含布尔型的成员**false**，在计数时也**不计**。

Index	Member
1	a
2	0
3	3
4	(null)
5	true
6	false



Index	是否计数
1	√
2	√
3	√
4	×
5	√
6	×



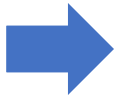
计数
4

9.3.1 计数、求和与均值

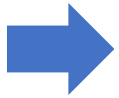
去重计数

- 去重计数是指统计序列中不重复成员的总个数，使用聚合函数A.icount()计算。
- 序列中的相同成员，在去重计数时只计算1次。
- 如果序列中包括null或者false，在去重计数时仍然不计。

Index	Member
1	Team Exercises
2	(null)
3	Party
4	Dinner
5	(null)
6	Party
7	Dinner
8	Party



Index	是否计数
1	√
2	×
3	√
4	√
5	×
6	○
7	○
8	○



去重计数
3

9.3.1 计数、求和与均值

【例题9.14】某部门准备组织团建活动，收到员工提议如下：Team Exercises, null, Party, Dinner, null, Party, Dinner, Party。其中如果无建议则填入空值，请问一共有多少种活动可供选择?

9.3.1 计数、求和与均值

【例题9.14】某部门准备组织团建活动，收到员工提议如下：Team Exercises, null, Party, Dinner, null, Party, Dinner, Party。其中如果无建议则填入空值，请问一共有多少种活动可供选择?

解题思路

由于数据中存在相同的项目，因此计数时需要使用去重计数函数**A.icount()**。

	A	B
1	[Team Exercises, null, Party, Dinner, null, Party, Dinner, Party]	/生成分数序列
2	=A1.icount()	/去重计数

A2	Value
	3

注意

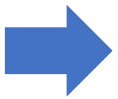
- 去重计数时，序列成员中的所有null和false都不会被计入。

9.3.1 计数、求和与均值

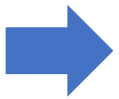
求和

- **求和**是指统计序列中所有成员的总和，使用聚合函数**A.sum()**计算。
- 在序列求和时，序列中**非数值**的成员都会被**忽略**，如null，字符串，时间日期，布尔数等。
- 如果序列中没有任何数值类型的成员，A.sum()的结果为null。

Index	Member
1	a
2	0
3	3
4	(null)
5	true
6	false



Index	是否求和
1	×
2	√
3	√
4	×
5	×
6	×



求和
3

9.3.1 计数、求和与均值

【例题9.15】已知几名学生某次测验的分数依次如下：66, null, 82, 96，其中有人因故未参加测试因此无成绩，请计算他们的总分和平均分，缺考学生不参加统计。

9.3.1 计数、求和与均值

【例题9.15】已知几名学生某次测验的分数依次如下：66, null, 82, 96，其中有人因故未参加测试因此无成绩，请计算他们的总分和平均分，缺考学生不参加统计。

解题思路

计算总分用求和函数**A.sum()**即可，计算平均分只需用总分/参考人数，参考人数用计数函数**A.count()**计算。

	A	B
1	[66, null, 82, 96]	/生成分数序列
2	=A1.sum()	/求和计算总分
3	=round(A2/A1.count(), 2)	/计算平均分，结果保留2位小数

A2	Value
	244

A3	Value
	81.33

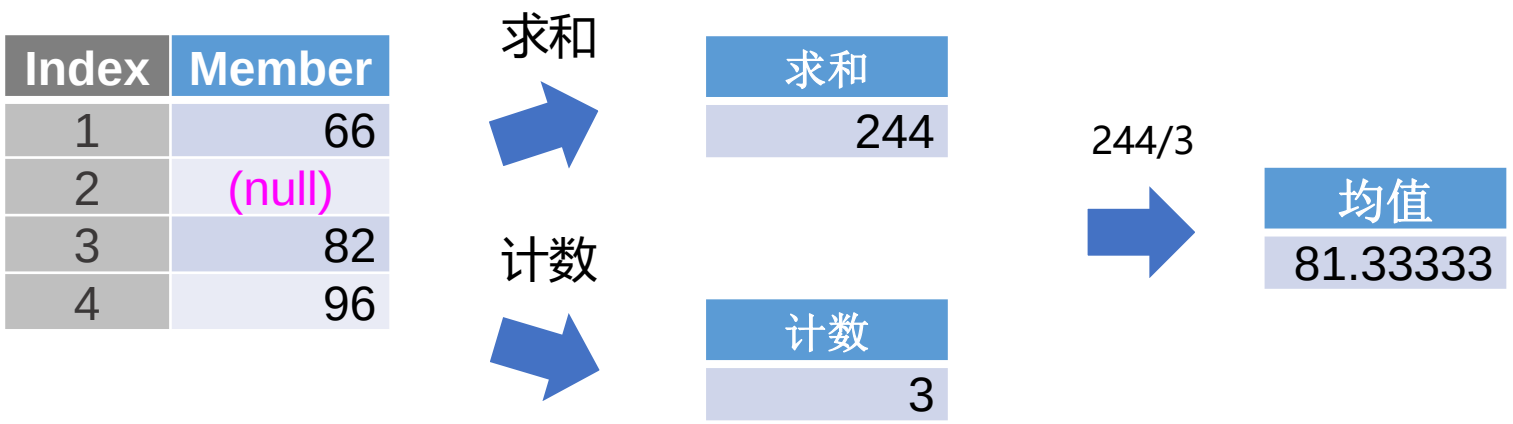
注意

- 计算均值也可以使用下面将要讲到的求均值函数**A.avg()**。A3中的代码也可以修改为**=round(A1.avg(), 2)**，结果是相同的，而且这种求均值的方法不需要先统计总分和计数。

9.3.1 计数、求和与均值

求均值

- 求均值是指统计序列中成员的**平均值**，使用聚合函数**A.avg()**计算。
- 在计算平均值时，序列中的**空值**是**不参与统计**的。
- 如果序列中除了空值，还含有其它**非数值类型的成员**，使用A.avg()计算时同样**不计**。



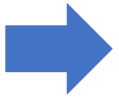
9.3.2 排名

排名

- 根据序列中成员的值按一定顺序排列，能够计算每个成员的排名。默认情况下**从小到大**排列，值最小的为第1名，最大的为最后一名。使用排名函数**A.ranks()**可以计算A中各个成员的排名序列。
- 执行排名计算时，序列中的成员的**类型必须能够相互比较**，如果有**空值**，将被认为**最小**。
- 如果需要计算某个值在序列中能够排在哪个名次，可以用**A.rank(y)**。

Index	Member
1	9.98
2	11.99
3	9.80
4	9.75
5	9.79
6	9.63
7	9.88
8	9.94

排名



Index	名次
1	7
2	8
3	4
4	2
5	3
6	1
7	5
8	6

9.3.2 排名

【例题9.16】某次短跑比赛中，参赛的8位选手成绩如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。求他们比赛的名次。如果某位选手的成绩是9.78秒，假如他参加比赛能够获得第几名?

9.3.2 排名

【例题9.16】某次短跑比赛中，参赛的8位选手成绩如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。求他们比赛的名次。如果某位选手的成绩是9.78秒，假如他参加比赛能够获得第几名?

解题思路

计算排名序列用**A.ranks()**函数，计算某个值在序列中的排名用**A.rank(y)**函数。

	A	B
1	[9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94]	/生成成绩序列
2	=A1.ranks()	/计算排名序列
3	=A1.rank(9.78)	/计算某个成绩的排名

A2	Index	Member
	1	7
	2	8
	3	4
	4	2
	5	3
	6	1
	7	5
	8	6

A3	Value
	3

注意

- 如果序列中存在null，计算排名时需要注意**排除空值的影响**。

9.3.2 排名

【例题9.17】某个部门中，员工的年龄如下：45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51。按照年龄从大到小，各员工的排名情况如何。如果并列的排名不占多余名次，排名情况如何？在数据统计中，为了消除并列排名产生的影响，经常需要计算统计排名，即如果有k个并列第n名，那么他们的名次记为(n+(n+1)+...(n+k-1))/k，如2个并列第2名，认为他们的排名是(2+3)/2=2.5。那么这个部门中各个员工的统计排名情况如何？

9.3.2 排名

【例题9.17】某个部门中，员工的年龄如下：45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51。按照年龄从大到小，各员工的排名情况如何。如果并列的排名不占多余名次，排名情况如何？在数据统计中，为了消除并列排名产生的影响，经常需要计算统计排名，即如果有k个并列第n名，那么他们的名次记为(n+(n+1)+...(n+k-1))/k，如2个并列第2名，认为他们的排名是(2+3)/2=2.5。那么这个部门中各个员工的统计排名情况如何？

解题思路

前面的赛跑成绩我们习惯于**从小到大**的**升序**排名，但是很多情况下还需要**从大到小**的**降序**排名。在A.ranks()和A.rank(y)函数中，可以使用各种选项，**@z**表示使用降序，**@i**表示并列排名不占名次，**@s**表示计算统计排名。其中@i和@s由于计算方案不同，因此不能同时使用。

	A	B
1	[45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51]	/生成年龄序列
2	=A1.ranks@z()	/计算年龄排名序列
3	=A1.ranks@zi()	/计算年龄排名序列，并列不占排名
4	=A1.ranks@zs()	/计算年龄统计排名序列

A2	Index	Member
	1	4
	2	5
	3	8
	4	2
	5	5
	6	7
	7	8
	8	8
	9	2
	10	8
	11	1

A3	Index	Member
	1	3
	2	4
	3	6
	4	2
	5	4
	6	5
	7	6
	8	6
	9	2
	10	6
	11	1

A4	Index	Member
	1	4.0
	2	5.5
	3	9.5
	4	2.5
	5	5.5
	6	7.0
	7	9.5
	8	9.5
	9	2.5
	10	9.5
	11	1.0

9.3.3 极值和中位数

最大/最小值

- 在序列的所有成员中，值最大的称为**最大值**，值最小的称为最小值。计算最大值和最小值可以分别使用聚合函数**A.max()**和**A.min()**。
- 计算最值时，序列中的空值，在统计时**不计**，既不会作为最大值，也不会作为最小值。

Index	Member
1	9.98
2	11.99
3	9.80
4	9.75
5	9.79
6	9.63
7	9.88
8	9.94

求最大值 → 最大值：11.99

求最小值 → 最小值：9.63

9.3.3 极值和中位数

【例题9.18】某个部门中，员工的年龄如下：45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51。部门中员工的最大年龄和最小年龄各是多少?

解题思路

求最大值或最小值直接使用聚合函数A.max()和A.min()即可。

	A	B
1	[45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51]	/生成年龄序列
2	=A1.max()	/计算最大年龄
3	=A1.min()	/计算最小年龄

A2	Value
	51

A3	Value
	21

9.3.3 极值和中位数

求中位数

- **中位数**又称**中值**，是序列中成员按顺序排列后，居于中间位置的数，使用**A.median()**获取。
- 在求中位数时，序列中各个成员的**数据类型**必须**可以进行比较**。
- 如果序列中的**成员个数为偶数**，那么最中间位置的数会有两个，此时返回它们的**平均值**。

Index	Member
1	7
2	3
3	5
4	2
5	6
6	1
7	4

排序

Member
1
2
3
4
5
6
7

中位数

中位数
4

9.3.3 极值和中位数

【例题9.19】某次短跑比赛中，参赛的8位选手成绩如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。求他们比赛成绩的中位数。

9.3.3 极值和中位数

【例题9.19】某次短跑比赛中，参赛的8位选手成绩如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。求他们比赛成绩的中位数。

解题思路

求中位数可以直接用聚合函数**A.median()**。

	A	B
1	[9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94]	/生成成绩序列
2	=A1.median()	/计算成绩中位数

A2	Value
	9.84

注意

- 共有8个成绩，排在中间的成绩有两个，第4位的9.80和第5位的9.88，因此求中位数得到的是它们的平均值9.84。

9.3.4 其它聚合函数

空值判断

- 在聚合计算中，可以查找**首个非空值**的**成员**，使用函数**A.ifn()**。
- 在聚合计算中，还可以查找**首个**既**非空值**也**非空字符串**的**成员**，使用函数**A.nvl()**。

Index	Member
1	(null)
2	(null)
3	
4	(null)
5	0
6	false

计数

Index	是否空值	是否空串
1	√	
2	√	
3	×	√
4	√	
5	×	×
6	×	×

A.ifn()

首个非空成员

A.nvl()

首个非空且非空串成员
0

9.3.4 其它聚合函数

逻辑与、逻辑或

- 在聚合计算中**计算逻辑与、逻辑或**，通常是用于布尔值序列，用来判断序列中的条件**是否全部成立**，以及**是否至少有一个条件成立**，可以分别使用**A.cand()**和**A.cor()**。
- 使用逻辑与、逻辑或的聚合运算的序列中，成员应该全部为布尔值。但如果序列中还有其它类型的成员，那么**空值视为false**，其它**非布尔值的成员视为true**。

Index	Member
1	(null)
2	(null)
3	
4	(null)
5	0
6	false

计数

Index	是否成立
1	×
2	×
3	√
4	×
5	√
6	×

A.cand()

是否全成立
false

A.cor()

是否有成立
true

9.4 循环计算

9.4.1 用A.(x)完成循环计算

A.(x)与循环计算

- 序列的简单循环计算除使用for A外，还可以使用**A.(x)**。
- **A.(x)**在执行时，会依次循环序列**A的每个成员**计算表达式**x**，**x**中可以用**~符号**引用当前成员参与计算，**A.(x)**将返回由结果构成的**新序列**。

	A	B
1	[1,2,3]	/原序列
2	=A1.(~*2)	/每个成员乘以2后，组成新序列返回

A1	Index	Member
	1	1
	2	2
	3	3

A2	Index	Member
	1	2
	2	4
	3	6

上面使用的A.(x)执行的结果，相当于右侧的代码，只是更为简单。

	A	B	C
1	[1,2,3]	[]	/原序列和结果序列
2	for A1	>B1=B1|(A2*2)	/循环原序列，在结果序列中添加一个计算值

9.4.1 用A.(x)完成循环计算

【例题9.20】已知4名学生某次测验的总分依次如下：232, 233, 224, 267。如果本次考试共包括3门科目，求每个人的平均分。

9.4.1 用A.(x)完成循环计算

【例题9.20】已知4名学生某次测验的总分依次如下：232, 233, 224, 267。如果本次考试共包括3门科目，求每个人的平均分。

解题思路

将测验成绩记录在序列中，使用 **A.(x)** 循环，根据每个学生的总分，分别计算出他们的平均分就好。

	A	B
1	[232, 233, 224, 267]	/生成总分序列
2	=A1.(round(~/3, 2))	/循环计算平均分，保留两位小数

A2	Index	Member
	1	77.33
	2	77.67
	3	74.67
	4	89.0

注意

- **A.(x)**相当于循环计算，针对**A**的每个成员计算**x**，并将结果构成新序列返回。**符号~**表示循环**A**时每次计算中使用的当前成员。

9.4.1 用A.(x)完成循环计算

【例题9.21】某种商品单价为9.0万元，5家公司预购数量依次为：25, 18, 33, 40, 9。现在销售公司做优惠活动，订单总金额超过100万元的部分打9折，请计算各家公司需付的总金额。

9.4.1 用A.(x)完成循环计算

【例题9.21】某种商品单价为9.0万元，5家公司预购数量依次为：25, 18, 33, 40, 9。现在销售公司做优惠活动，订单总金额超过100万元的部分打9折，请计算各家公司需付的总金额。

解题思路

先使用 **A.(x)** 循环计算打折前的总价，再使用**A.(~=x)**的循环方式修改总价，计算打折后的金额。

	A	B	C
1	[25, 18, 33, 40, 9]	=A1.(~*9.0)	/生成销量序列，计算打折前总价
2	=B1.()	=B1.(~)	/循环计算打折前的总价
3	=B1.(if(~>100, ~=(~-100)*0.9+100))		/对于总价大于100万元的，打折处理

B1	Index	Member
	1	212.5
	2	155.8
	3	277.3
	4	334.0
	5	81.0

A2	Index	Member
	1	212.5
	2	155.8
	3	277.3
	4	334.0
	5	81.0

B2	Index	Member
	1	225.0
	2	162.0
	3	297.0
	4	360.0
	5	81.0

A3	Index	Member
	1	212.5
	2	155.8
	3	277.3
	4	334.0
	5	(null)

注意

- 在不写表达式x时，**A.()**会返回序列**A本身**。
- 从结果中可以看到，B1和A2中的序列完全相同，因为A2中返回的结果就是B1本身。
- B2中的结果仍然保持了打折前的数据，这是由于x不是赋值语句时，**A.(x)**返回的结果是个**新序列**，与A无关。
- A3在循环计算时只修改了超过100万元的金额，所以结果的最后一个成员为null.

9.4.1 用A.(x)完成循环计算

【例题9.22】计算1~6的阶乘

9.4.1 用A.(x)完成循环计算

【例题9.22】计算1~6的阶乘

解题思路

在A.(x)使用的表达式x中，可以用符号 # 在循环中引用当前成员的序号，用符号 ~[-i] 在循环中引用当前成员之前的第 i 个成员，也就是说~[-1]表示引用A中前一个成员。在SPL中除了相对引用一个成员，还可以用符号~[a, b]相对引用一个区间内的成员，，如~[-3,-1]表示引用前3个成员。

	A	B
1	=[0]*6	/A*n相当于n个A的合集，[0]*6会得到[0,0,0,0,0,0]，用来记录结果
2	=A1.(~=if(#>1, #*~[-1], 1))	/循环计算，1的阶乘为1，后面计算n!等于n*(n-1)!

上面的代码还可以进一步简化：

	A	B
1	=6.(~=if(#>1, #*~[-1], 1))	/循环计算，1的阶乘为1，之后n!等于n*(n-1)!

A1	Index	Member
	1	1
	2	2
	3	6
	4	24
	5	120
	6	720

注意

- n.(x)相当于计算to(n).(x)，而to(n)会生成1~n构成的序列。

9.4.1 用A.(x)完成循环计算

【例题9.22】计算1~6的阶乘。

注意

- 在前面的解法中，在后面的计算中需要使用前面的结果，比如数列中已有的结果。遇到这样的问题，可以在循环中使用迭代函数**A.(iterate(x,a))**。迭代函数在循环表达式**x**中，可以用**~~**符号调用迭代值，迭代值就是每一次循环中计算表达式的结果；循环执行前，迭代值的初始值为**a**。

	A	B
1	=6.(iterate(~~*~, 1))	/迭代计算，阶乘的初始值为1，每次将迭代结果乘以当前成员的值

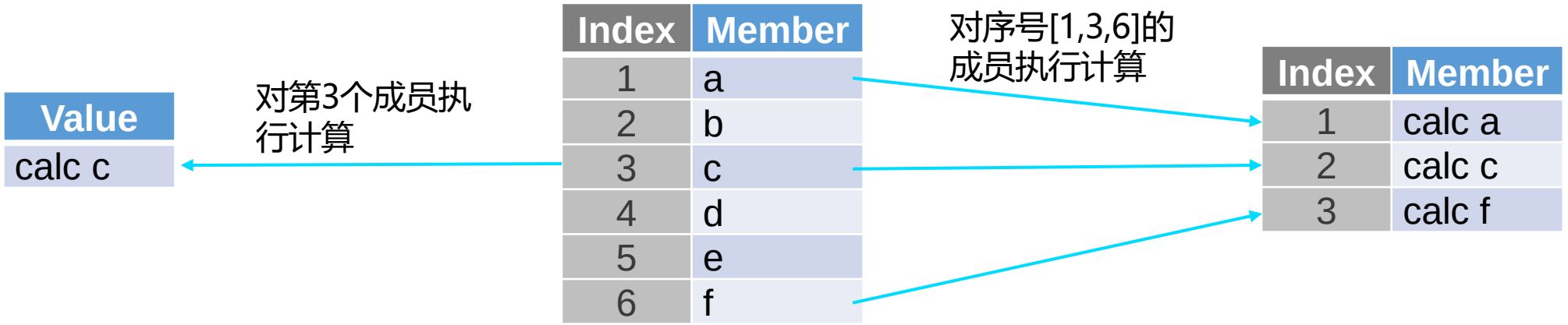
A1	Index	Member
	1	1
	2	2
	3	6
	4	24
	5	120
	6	720

有关迭代函数的内容，将在第12章第3节中详细讲解。

9.4.2 定位计算

定位计算

- A.(x)会循环针对序列中的**每个成员**进行计算。有的时候，只需要对A中**指定序号**的部分成员执行计算就可以了，这种针对指定位置执行的计算称为**定位计算**。
- 定位计算使用A.calc(k, x)或者A.calc(p, x)，分别对A中的第k个成员执行计算，或者对A中**序号**在数列p中的各个成员执行计算。



9.4.2 定位计算

【例题9.23】根据例题9.16中得到的1~6阶乘，得到指定数的阶乘等式，如：5! = 120。

9.4.2 定位计算

【例题9.23】根据例题9.16中得到的1~6阶乘，得到指定数的阶乘等式，如：5! = 120。

解题思路

根据上一个例题的结果，用**A.calc(k, x)** 定位计算获得阶乘等式。

	A	B
1	=6.(~=if(#>1, #*~[-1], 1))	/循环计算，1的阶乘为1，之后n!等于n*(n-1)!
2	=A1.calc(5, #/"! = "/~)	/定位计算，输出5的阶乘等式

A2	Value
	5! = 120

注意

- 定位计算时，只针对指定位置的成员执行计算，而不必循环整个序列。

9.4.2 定位计算

【例题9.24】交易所某支股票在一周内的收盘价依次如下：8.12, 8.22, 9.03, 9.53, 9.32。求每天的收盘价涨幅，首日不计。

9.4.2 定位计算

【例题9.24】交易所某支股票在一周内的收盘价依次如下：8.12, 8.22, 9.03, 9.53, 9.32。求每天的收盘价涨幅，首日不计。

解题思路

计算涨幅需要和前一天的收盘价作比较，排除首日后可以用**A.calc(p, x)**来定位计算。

	A	B
1	[8.12, 8.22, 9.03, 9.53, 9.32]	/生成收盘价序列
2	=A1.calc(to(2,A1.len()), round((~-~[-1])/~[-1], 3))	/定位计算涨幅，保留3位小数

A2	Index	Member
	1	0.012
	2	0.099
	3	0.055
	4	-0.022

注意

- 用**to(2, A1.len())**计算2~n的定位数列，其中**A1.len()** 可以返回A1的序列长度。
- 定位计算的表达式中，仍然可以相对引用前一个成员执行计算。

9.4.3 循环函数

循环函数

- 循环函数的通用格式是**A.f(x)**，可以针对序列的**每个成员**进行计算，可以将常用模式的一些循环语句用简单的函数表达出来。
- 循环函数的本质，是对循环计算的一种简化。所以每一种循环函数，都可以找到对应的循环计算方法。
- 前面介绍的聚合函数，实际上也可以看做无参数的循环函数，而作为循环函数使用时，还可以添加参数**x**使用。除此以外，循环函数还有返回成员序号的**定位函数**以及选出全部或部分成员执行计算的**选出函数**等，这两种类型的函数将在后面分别用一节介绍。本小节将主要讲解其它的循环函数，包括**A.run()，A.rank(y, x)，A.median(k:n, x)**等。

9.4.3 循环函数

【例题9.25】某个部门中，员工的出生日期如下： 1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。求部门中员工的平均年龄。

9.4.3 循环函数

【例题9.25】某个部门中，员工的出生日期如下： 1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。求部门中员工的平均年龄。

解题思路

年龄需要根据生日用age(d)函数计算，平均年龄可以直接用A.avg(x)形式的聚合函数来计算。

	A	B
1	[1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13]	/生成生日序列
2	=round(A1.avg(age(~)), 2)	/定位平均年龄，保留2位小数

A2	Value
	38.33

注意

- 计算**A.f(x)**，相当于先计算**A.(x)**得到序列**B**，再计算**B.f()**。
- 具体到本例，相当于先计算**A1.(age(~))**得到年龄序列，再用A.avg()计算平均年龄。各种聚集函数都可以用A.f(x)的格式使用，如**A.sum(x)**，**A.count(x)等**。

9.4.3 循环函数

A.run(x)

- A.run(x)循环在A中执行表达式x，最后仍然返回A。
- A.run(x)中的循环表达式x需要改变A，一般为赋值类语句，这与A.(x)不同。
- A.run(x)返回的仍然是A，所以经常用在执行格中；而A.(x)返回的是循环计算结果构成的新序列。

调整价格：超过100万元的部分打9折

Index	Member
1	225.0
2	162.0
3	297.0
4	360.0
5	81.0

A.run(x)

Index	Member
1	212.5
2	155.8
3	277.3
4	334.0
5	81.0

A.(x)

Index	Member
1	212.5
2	155.8
3	277.3
4	334.0
5	(null)

【例题9.26】某种商品单价为9.0万元，5家公司预购数量依次为：25, 18, 33, 40, 9。现在销售公司做优惠活动，订单总金额超过100万元的部分打9折，请用A.run(x)计算各家公司需付的总金额。

9.4.3 循环函数

A.run(x)

【例题9.26】某种商品单价为9.0万元，5家公司预购数量依次为：25, 18, 33, 40, 9。现在销售公司做优惠活动，订单总金额超过100万元的部分打9折，请用A.run(x)计算各家公司需付的总金额。

解题思路

先使用 **A.(x)** 循环计算打折前的总价，再使用**A.run(~=x)**的循环方式修改。

	A	B	C
1	[25, 18, 33, 40, 9]		/生成销量序列，计算打折前总价
2	=A1.(~*9.0)	=A2.(~)	/循环计算打折前总价
3	=A2.run(if(~>100, ~=(~-100)*0.9+100))	=B2.(if(~>100, ~=(~-100)*0.9+100))	/对于总价大于100万元的，打折处理

A2	Index	Member
	1	212.5
	2	155.8
	3	277.3
	4	334.0
	5	81.0

B2	Index	Member
	1	212.5
	2	155.8
	3	277.3
	4	334.0
	5	81.0

A3	Index	Member
	1	212.5
	2	155.8
	3	277.3
	4	334.0
	5	81.0

B3	Index	Member
	1	212.5
	2	155.8
	3	277.3
	4	334.0
	5	(null)

注意

- 为了比较A.run(x)和A.(x)的区别，在B2中“复制”了A2中打折前的总价序列。
- A3中用A.run(x)函数修改了A2中的序列，最后的结果中A3与A2是完全相同。
- B3中用A.(x)函数修改了B2中的序列，同样完成了打折的修改，所以B2被修改后和A2是相同的。但是由于B3中只修改了超过100万的数据，因此B3中最后一个成员是null。

其它循环函数

- 循环函数的基本格式都是**A.f(x)**，都相当于**先计算A.(x)**，再在结果序列B中**计算B.f()**。
- **A.rank(y, x)**与聚合函数A.rank(y)的区别就是并不是计算**y**在A中的排名，而是计算**y**在**A.(x)**中的排名。
- **A.median(k:n, x)**位于序列中k/n位置的成员，同样是先计算**A.(x)**，再在结果中计算分位数。

【例题9.27】某个部门中，员工的出生日期如下： 1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。现在准备安排不少于三分之一的员工值班，如果想优先安排年轻员工值班的话，值班员工的年龄设为不大于多少合适?

9.4.3 循环函数

【例题9.27】某个部门中，员工的出生日期如下： 1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。现在准备安排不少于三分之一的员工值班，如果想优先安排年轻员工值班的话，值班员工的年龄设为不大于多少合适?

解题思路

使用 **A.median(k:n, x)** 计算位于年龄升序排序时，位于**k/n**位置的年龄值。在**A.median(k:n)**函数先将**A**中的数据分成n组，不能等分时，前面组中成员可能比后面组中成员多1个，因此能够满足最低选出**k/n**员工值班的条件。当数据正好能够分为**n**组时，中位数将为**k/n**位置两侧的两个成员的均值，此时将正好有**k/n**员工满足值班条件。

	A	B
1	[1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13]	/生成生日序列
2	=A1.median(1:3, age(~))	/求年龄界限

A2	Value
	38.33

9.5 定位函数

9.5 定位函数

定位函数

- 在集合中找到指定的一个或多个成员的位置，这样的运算称为**定位**。
- 执行定位的函数在执行时，默认情况下会从序列中的第1个成员开始**从前向后**查找，直到找到**首个**满足条件的成员，返回它的**序号**。
- 使用定位函数时，如果添加**@z**选项，将反向从后向前查找；如果添加**@a**选项，将会返回**所有**满足条件成员的**序号数列**。
- 如果序列中的成员是**有序**的，可以添加**@b**选项，在查找时使用**二分法**以大幅提高效率。
- 常用定位函数包括**A.pos(x), A.pselect(x), A.pmax(x), A.pmin(x), A.pseg(x), A.psort(x), A.contain(x), A.ptop()**等。

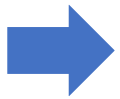
9.5.1 查找指定成员的位置

查找指定成员的位置

- 在序列中查找某个指定成员x的位置，可以使用A.pos(x)函数。
- 计算A.pos(x)时，如果在A中能够找到x，则返回首个x的序号；如果找不到则返回null。
- 如果添加@n选项，则在无法找到x时返回A.len()+1。

Index	Member
1	a
2	b
3	c
4	d
5	d
6	c

查找c的位置



序号
3

9.5.1 查找指定成员的位置

【例题9.28】某个部门中，员工的年龄如下：45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51。请问第几位员工25岁？第几位员工50岁？哪几位员工21岁？

9.5.1 查找指定成员的位置

【例题9.28】某个部门中，员工的年龄如下：45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51。请问第几位员工25岁？第几位员工50岁？哪几位员工21岁？

解题思路

使用 **A.pos(x)** 查找指定成员的序号，如果需要返回序列中所有值为x成员的序号，则需要添加@a选项。

	A	B
1	[45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51]	/生成年龄序列
2	=A1.pos(25)	/求25岁员工序号
3	=A1.pos(50)	/求50岁员工序号
4	=A1.pos@a(21)	/求所有21岁员工序号

A2	Value
	6

A3	Value
	(null)

A4	Index	Member
	1	3
	2	7
	3	8
	4	10

注意

- A3中无法在序列中找到50，所以返回null。此时如果将A3中的表达式改为=A1.pos@n(50)，由于A1中共11个成员，结果会返回12。
- A4中添加了@a选项，返回了所有值为21的成员的序号所构成的数列。在计算A.pos@a(x)时，如果只找到1个x，也会返回1个成员的序列；如果无法找到x，会返回空序列。

9.5.2 查找满足条件成员的位置

查找满足条件成员的位置

- 在序列中满足条件x的成员所在位置，可以使用A.pselect(x)函数。
- 计算A.pselect(x)时，将返回首个满足x的成员序号；如果找不到则返回null。
- 如果添加@n选项，则在所有成员都无法满足x时返回A.len()+1。

Index	Member
1	1
2	2
3	6
4	24
5	120
6	720

查找大于100
的成员位置

序号
5

9.5.2 查找满足条件成员的位置

【例题9.29】某个部门中，员工的年龄如下：45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51。请问年龄超过了40岁的员工序号是哪些?

9.5.2 查找满足条件成员的位置

【例题9.29】某个部门中，员工的年龄如下：45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51。请问年龄超过了40岁的员工序号是哪些?

解题思路

根据条件在序列中查找成员位置，使用 **A.pselect(x)** 函数，如果需要返回序列中所有满足条件成员的序号，则需要添加**@a**选项。

	A	B
1	[45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51]	/生成年龄序列
2	=A1.pselect(~>40)	/求40岁以上员工序号
3	=A1.pselect@a(~>40)	/求40岁以上所有员工序号

A2	Value
	1

A3	Index	Member
	1	1
	2	4
	3	9
	4	11

注意

- A2中只返回了**第一位**超过40岁员工的序号。
- A3中添加了**@a**选项，返回了所有超过40岁员工的序号构成的序列，A3中的才是题目所需的结果。

9.5.3 查找最大/最小值的位置

查找极值所在的位置

- 在序列中极值所在位置，可以使用**A.pmax()/A.pmin()**函数。
- 如果添加**x**参数，**A.pmax(x)** 相当于**A.(x).pmax()**，先计算出**A.(x)**再返回最大值所在位置。**pmin()**同理 。
- 默认从前往后找，使用**@z**选项可以从后往前找。如果有多个相同的极值，默认返回第1个，使用**@a**选项可以返回所有。

Index	Member
1	9.98
2	11.99
3	9.80
4	9.75
5	9.79
6	9.63
7	9.88
8	9.94

最大值的位置 → 序号：2

最小值的位置 → 序号：6

9.5.3 查找最大/最小值的位置

【例题9.30】某次短跑比赛中，1~8跑道参赛的选手成绩依次如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。请问哪条跑道的选手成绩最好，哪条跑道的选手成绩最差?

9.5.3 查找最大/最小值的位置

【例题9.30】某次短跑比赛中，1~8跑道参赛的选手成绩依次如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。请问哪条跑道的选手成绩最好，哪条跑道的选手成绩最差?

解题思路

问题是求最大值和最小值的位置，需要使用A.pmax()和A.pmin()函数，注意短跑比赛成绩最好为最小值，成绩最差为最大值。

	A	B
1	[9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94]	/生成成绩序列
2	=A1.pmin()	/成绩最好选手跑道
3	=A1.pmax()	/成绩最差选手跑道

A2	Value
	6

A3	Value
	2

9.5.3 查找最大/最小值的位置

【例题9.31】某个部门中，员工的出生日期如下： 1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。请问第几位员工的年龄最大?

9.5.3 查找最大/最小值的位置

【例题9.31】某个部门中，员工的出生日期如下： 1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。请问第几位员工的年龄最大?

解题思路

需要查找的是最大年龄的员工序号，但是年龄需要计算得到，可以用A.pmax(x)计算。

	A	B
1	[1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13]	/生成生日序列
2	=A1.pmax(age(~))	/求最大年龄的员工序号

A2	Value
	3

注意

- 这道题目也可以转换思路：年龄最大的员工出生日期最早，用**A.pmin()**计算。

9.5.4 整体定位

整体定位

- **整体定位函数**是指需要针对序列中所有成员执行定位一些定位函数。这些函数本身就需在所有成员中查找，因此不支持@a选项，同时也不支持@z选项，无法逆向查找。
- 整体定位包括下面的一些函数：

  在序列中同时查找多个成员的定位函数A.pos(p)，其中p为多个成员构成的序列；

  查找序列中是否包含某些成员的A.contain(x_i,...)函数；

  将序列从小到大（或从大到小）排列后，计算各个成员的序号数列的A.psort(x)函数；

  查找序列中前几名成员的序号数列的A.ptop(n, x)函数；

  当A中成员有序，用来表示一组区间划分界限时，查找数据所在区间的A.pseg(x)函数。

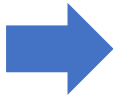
9.5.4 整体定位

多成员定位

- A.pos(p)函数用来查找多个成员在序列中的位置数列。
- 和前面的A.pos(x)不太相同，在执行A.pos(p)时，对于p中的每个成员，结果只会返回找到的第1个位置，**不能使用@a选项**。
- 和单独查找p中的每个成员不同，如果p中至少1个成员无法找到，都会认为**p无法在A中定位**，A.pos(p)函数将返回null。

Index	Member
1	a
2	b
3	c
4	d
5	d
6	c

查找c和a的位置



Index	Member
1	3
2	1

9.5.4 整体定位

【例题9.32】某美术学校每期开设不同的公开课，各期将要陆续设置的课程下： Watercolor, Cartoon, Chromatics, Sketch, Watercolor, Caricature, Oil Painting, Sketch。如果A学生想学习素描(Sketch)和色彩(Chromatics)，他可以如何选择？如果B学生想学习漫画(Caricature)和摄影(Photography)，他该如何选择？如果C学生想先学习素描再学习水彩(Watercolor)，他可以如何选择？如果C还想连续学习这两期课程，可以做到吗？

9.5.4 整体定位

【例题9.32】某美术学校每期开设不同的公开课，各期将要陆续设置的课程下： Watercolor, Cartoon, Chromatics, Sketch, Watercolor, Caricature, Oil Painting, Sketch。如果A学生想学习素描(Sketch)和色彩(Chromatics)，他可以如何选择？如果B学生想学习漫画(Caricature)和摄影(Photography)，他该如何选择？如果C学生想先学习素描再学习水彩(Watercolor)，他可以如何选择？如果C还想连续学习这两期课程，可以做到吗？

解题思路

在序列中查找多个成员的位置，可以使用**A.pos(p)**函数解决。如果需要**p**中的成员在**A**中是**连续**的，可以添加**@c**选项。如果在A中查找p的成员时，需要从前向后查找，也就是说p中在后面的成员在A中的位置也必须在后面，可以添加@i选项，此时返回的序号数列必须是**单调递增**的。

	A	B
1	[Watercolor, Cartoon, Chromatics, Sketch, Watercolor, Caricature, Oil Painting, Sketch]	/生成课程序列
2	=A1.pos(["Sketch", "Chromatics"])	/学生A可选哪些期课程
3	=A1.pos(["Caricature", "Photography"])	/学生B可选哪些期课程
4	=A1.pos@i(["Sketch", "Watercolor"])	/学生A可选哪些期课程
5	=A1.pos@c(["Sketch", "Watercolor"])	/学生B可选哪些期课程

A4	Index	Member
	1	4
	2	3

A3	Value
	(null)

A4	Index	Member
	1	4
	2	5

A5	Value
	4

查找多个成员时，每个成员会返回找到的**首个**序号。

即使p中有1个成员无法找到，也会返回**null**。

使用**@i**选项时，在A中将**单向**查找，返回数列一定是**单调递增**的。

使用**@c**选项时，要求p中的成员是**连续**出现在A中的，会返回找到的首个成员的序号，而不是数列。

判断是否包含多个成员

- A.contain(x_i,...)函数用来查找多个成员是否都在序列A中，返回结果为布尔值。
- A.contain(x_i,...)在判断序列中是否包含指定成员时，必须包含所有x_i,...才会返回true；参数中，如果存在n个相同的成员，那么判断时，A中**只需要存在1个**就会认为满足包含条件。

Index	Member
1	a
2	b
3	c
4	d
5	d
6	c

是否包含c和a

是否包含
true

9.5.4 整体定位

【例题9.33】某美术学校每期开设不同的公开课，各期将要陆续设置的课程下： Watercolor, Cartoon, Chromatics, Sketch, Watercolor, Caricature, Oil Painting, Sketch。如果某学生想学习素描(Sketch)、色彩(Chromatics)和水彩(Watercolor)，他可以选择在该学校学习吗?

9.5.4 整体定位

【例题9.33】某美术学校每期开设不同的公开课，各期将要陆续设置的课程下： Watercolor, Cartoon, Chromatics, Sketch, Watercolor, Caricature, Oil Painting, Sketch。如果某学生想学习素描(Sketch)、色彩(Chromatics)和水彩(Watercolor)，他可以选择在该学校学习吗?

解题思路

判断序列中是否包含多个成员，可以用**A.contain(x_i)**函数，也可以判断**A.pos(p)**函数的结果是否为空值。

	A	B
1	[Watercolor, Cartoon, Chromatics, Sketch, Watercolor, Caricature, Oil Painting, Sketch]	/生成课程序列
2	=A1.contain("Sketch", "Chromatics", "Watercolor")	/想学的课程是否都开设了
3	=A1.pos(["Sketch", "Chromatics", "Watercolor"]) != null	/用pos函数是否为空值判断

A2	Value
	true

A3	Value
	true

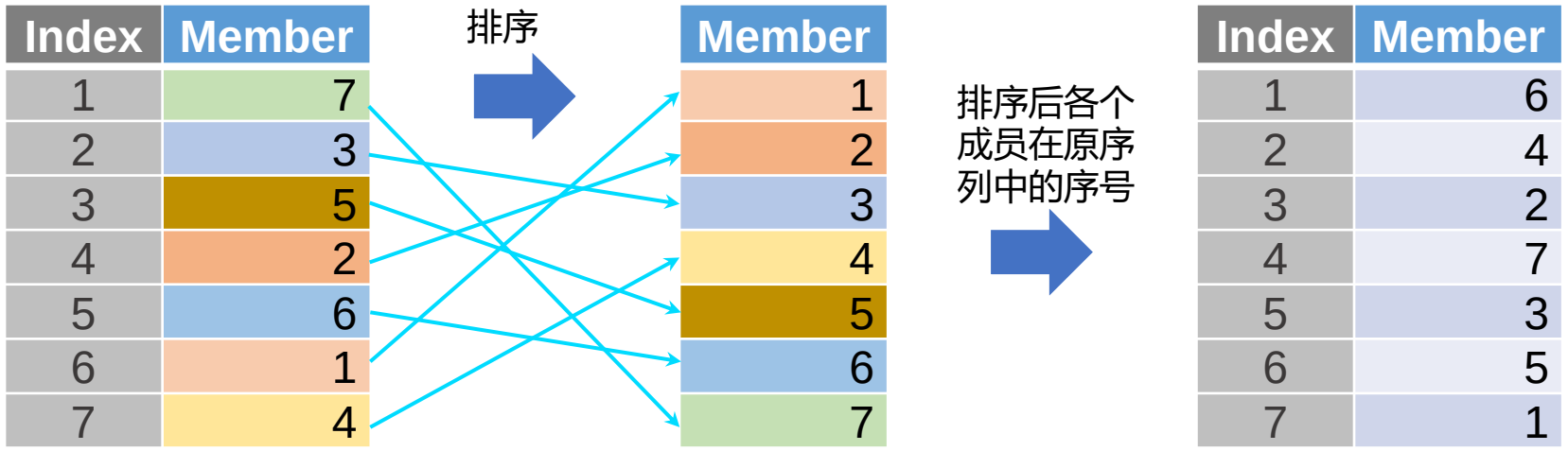
注意

- A2和A3中的结果是完全相同的。 **A.contain(x_i)**函数的参数不是序列，而是依次输入各个成员值，而且这个函数效果比较单一，不支持单向查找、连续查找等功能。

9.5.4 整体定位

排序定位

- A.psort()函数将返回A升序排序后，各个成员的初始序号构成的数列。
- 如果添加x参数，A.psort(x)相当于A.(x).psort()，先计算出A.(x)再返回它的排序定位数列。
- A.psort(x)函数可以添加选项@z， A.psort@z(x)将返回逆序排序(降序)的定位数列。



Index	Member
1	7
2	3
3	5
4	2
5	6
6	1
7	4

Member
1
2
3
4
5
6
7

Index	Member
1	6
2	4
3	2
4	7
5	3
6	5
7	1

9.5.4 整体定位

【例题9.34】某次短跑比赛中，1~8跑道参赛的选手成绩依次如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。请问各个名次的选手所在跑道?

9.5.4 整体定位

【例题9.34】某次短跑比赛中，1~8跑道参赛的选手成绩依次如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。请问各个名次的选手所在跑道?

解题思路

这里是求数据排序后的原始序号，应该使用A.psort()函数。

	A	B
1	[9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94]	/生成成绩序列
2	=A1.psort()	/各名次选手所在跑道
3	=A1.psort@z()	/逆序获取各名次选手所在跑道

A2	Index	Member
	1	6
	2	4
	3	5
	4	3
	5	7
	6	8
	7	1
	8	2

A3	Index	Member
	1	2
	2	1
	3	8
	4	7
	5	3
	6	5
	7	4
	8	6

注意

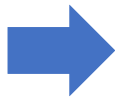
▶A2和A3中结果的顺序正好相反。解决排序问题时需要思考应该选择哪种顺序合适，作为短跑比赛成绩，排列名次时应该采取升序，A2中的代码才是正确的。

9.5.4 整体定位

前n名定位

- A.ptop(n)函数用来查找序列升序排序后，前n个成员的序号数列。
- 如果添加x参数，A.ptop(n, x)相当于A.(x).ptop(n)，先计算出A.(x)再返回它的前n名序号数列。
- A.ptop(n, x)函数并不支持@z选项，无法直接使用降序排序查找前n名，但是可以在表达式x前或者n前添加负号实现降序排序的效果。

Index	Member
1	9.98
2	11.99
3	9.80
4	9.75
5	9.79
6	9.63
7	9.88
8	9.94



Index	序号
1	6
2	4
3	5

9.5.4 整体定位

【例题9.35】某次短跑比赛中，1~8跑道参赛的选手成绩依次如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。请问前3名选手所在的跑道?

9.5.4 整体定位

【例题9.35】某次短跑比赛中，1~8跑道参赛的选手成绩依次如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。请问前3名选手所在的跑道?

解题思路

这里是求数据排序后的原始序号，应该使用A.psort()函数。

	A	B
1	[9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94]	/生成成绩序列
2	=A1.ptop(3)	/各名次选手所在跑道
3	=A1.psort().to(3)	/逆序获取各名次选手所在跑道

A2	Index	Member
	1	6
	2	4
	3	5

A3	Index	Member
	1	6
	2	4
	3	5

注意

- A2和A3中的结果是相同的。如果取前n名，确实可以将所有成员排序后再取前n名的位置，但是将所有成员排序的运算量会比较大，特别是数据量较多的时候，所以A2中的算法更合适，效率更高。

【例题9.36】某个部门中，员工的出生日期如下： 1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。请问年龄最大的3位员工的序号？年龄最小的员工序号呢?

9.5.4 整体定位

【例题9.36】某个部门中，员工的出生日期如下： 1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。请问年龄最大的3位员工的序号？年龄最小的员工序号呢?

解题思路

需要查找“年龄最大的3位员工”，因此需要使用**降序**排序，需要**在排序表达式或n前加负号**。查找年龄最小的员工序号，可以使用最值定位函数A.pmin(x)，也可以用A.ptop(n,x)取前1名的序号作为结果。

	A	B	C
1	[1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13]		/生成生日序列
2	=A1.ptop(-3, age(~))	=A1.ptop(1, age(~))	/年龄最大的3位员工序号和最小的员工序号
3	=A1.pmin(age(~))	=A1.ptop@1(1, age(~))	/最小的员工序号

A2	Index	Member
	1	3
	2	1
	3	2

B2	Index	Member
	1	6

A3	Value
	6

B3	Value
	6

注意

- A2中的表达式也可以写为**=A1.ptop(3, -age(~))**，效果是相同的。
- B2和A3中的结果不同，**A.ptop(n, x)**返回的结果是**数列**，即使定位“第1名”也会返回一个值构成的数列；而**A.pmin(x)**返回的是**序号**。如果添加选项，使用**A.ptop@1(n, x)**查找**第1名或最后一名**，可以不返回数列同样返回序号，如B3所示。

9.5.4 整体定位

区间定位

- 当A中成员有序，用来表示一组区间划分界限时，可以用A.pseg(x)函数查询x位于哪个区间。
- 使用A.pseg(x) 定位区间时，每个区间默认都会采取左闭右开区间，可以添加选项@r采用左开右闭的区间规则。

Index	Member
1	0
2	18.5
3	24
4	28

查找23.2的分区号

分区序号
2

默认情况下，Apseg(x)定位区间时采取左闭右开区间，在这个例子中，相当于设置了以下四个区间：[0, 18.5)，[18.5, 24)，[24, 28)，[28, +∞)。18.5≤23.2<24，所以23.2被划分到第2个区间。如果x<0，则返回的区间号为0。

【例题9.37】体质指数BMI (Body Mass Index) 常用来衡量人的健康状况。BMI=体重/身高^2，体重单位为kg，身高单位为m。BMI<18.5过瘦，18.5≤BMI<24身材正常，24≤BMI<28超重，BMI≥28肥胖。如果某人身高1.78m，体重80.2kg，他的身材评价如何?

9.5.4 整体定位

【例题9.37】体质指数BMI (Body Mass Index) 常用来衡量人的健康状况。BMI=体重/身高^2，体重单位为kg，身高单位为m。BMI<18.5过瘦，18.5≤BMI<24身材正常，24≤BMI<28超重，BMI≥28肥胖。如果某人身高1.78m，体重80.2kg，他的身材评价如何?

解题思路

先计算出BMI，再用**A.pseg(x)**函数判断BMI的分区号，最后用序号从评价中获得身材评语。

	A	B	C
1	[0,18.5,24,28]	[Lean, Fit, Overweight, Obese]	/生成BMI区间序列和评价序列
2	=80.2/1.78/1.78		/计算BMI
3	=A1.pseg(A2)	=B1(A3)	/查找BMI所在区间，获取评价

A2	Value
	25.31

A3	Value
	3

B3	Value
	Overweight

9.6 选出计算

9.6 选出计算

选出计算

- 在集合中根据条件找到一个或多个成员，这样的运算称为**选出计算**。
- 选出计算在默认情况下会在序列中**从前向后**查找，直到找到**所有**满足条件的成员，大部分选出函数会返回这些成员构成的**新序列**，也会有一些选出函数默认会返回**单值**。
- 使用选出函数时，如果添加**@z**选项，将反向从后向前查找。此时得到的结果，序列中成员的顺序是和无选项时不同的。
- 如果序列中的成员是**有序**的，可以添加**@b**选项，在查询时使用**二分法**以大幅提高效率。
- 常用选出函数包括**A.select(x), A.maxp(x), A.minp(x), A.sort(x), A.top(x)**等。

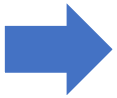
9.6.1 查找满足条件的成员

查找满足条件的成员

- 在序列中找出**所有**满足条件**x**的成员，可以使用**A.select(x)**函数。
- 计算**A.select(x)**时，将返回**所有**满足**x**的成员构成的**序列**；如果找不到则返回**空序列。**
- 如果添加**@1**选项，则只会找到**有第一个满足条件x的成员**，并返回该成员**单值**。如果使用**A.select@1(x)**时没能找到记录，将返回**null**，而不是空序列。

Index	Member
1	1
2	2
3	6
4	24
5	120
6	720

查找大于100的成员



Index	Member
1	120
2	720

9.6.1 查找满足条件的成员

【例题9.38】某个部门中，员工的年龄如下：45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51。请问哪些年龄超过了40岁？分别用从前向后和从后向前的顺序查找结果。

9.6.1 查找满足条件的成员

【例题9.38】某个部门中，员工的年龄如下：45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51。请问哪些年龄超过了40岁？分别用从前向后和从后向前的顺序查找结果。

解题思路

根据条件在序列中查找成员，使用 **A.select(x)** 函数，如果需要从后向前查找，则需要添加**@z**选项。

	A	B
1	[45, 33, 21, 49, 33, 25, 21, 21, 49, 21, 51]	/生成年龄序列
2	=A1.select(~>40)	/求40岁以上的年龄有哪些
3	=A1.select@z(~>40)	/反向查找40岁以上的年龄有哪些

A2	Index	Member
	1	45
	2	49
	3	49
	4	51

A3	Index	Member
	1	51
	2	49
	3	49
	4	45

注意

- A2和A3结果的序列中，成员的顺序是相反的，这和查找时使用的**顺序**有关。
- 使用**A.select(x)**函数选出成员时，即使只找到1个成员，也会返回1个成员的**序列**；在没找到符合条件的成员时，也会返回**空序列**。

9.6.2 最大/最小值输出

查找最大/最小值

- 在序列中查找最大值或最小值，可以使用**A.maxp()**或**A.minp()**函数。
- 如果需要先针对序列计算表达式x，再查找最大值或最小值所**对应的A中的成员**，使用**A.maxp(x)**或**A.minp(x)**函数。如果需要返回最大值或最小值本身，需要使用**A.max(x)**和**A.min(x)**。
- **A.maxp()**或**A.minp()**函数默认情况下之后返回一个单值，如果需要返回所有最大/最小值，可以添加**@a**选项。

Index	Member
1	9.98
2	11.99
3	9.80
4	9.75
5	9.79
6	9.63
7	9.88
8	9.94

最大值 →

最大值
11.99

最小值 →

最小值
9.63

9.6.2 最大/最小值输出

【例题9.39】某次短跑比赛中，1~8跑道参赛的选手成绩依次如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。请问哪个成绩最好，哪个成绩最差?

9.6.2 最大/最小值输出

【例题9.39】某次短跑比赛中，1~8跑道参赛的选手成绩依次如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。请问哪个成绩最好，哪个成绩最差?

解题思路

问题是求最大值和最小值，需要使用**A.maxp()**和**A.minp()**函数，注意短跑比赛成绩最好为最小值，成绩最差为最大值。

	A	B
1	[9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94]	/生成成绩序列
2	=A1.minp()	/最好成绩
3	=A1.maxp()	/最差成绩

A2	Value
	9.63

A3	Value
	11.99

9.6.2 最大/最小值输出

【例题9.40】某个部门中，员工的出生日期如下： 1974-11-20, 1995-04-11, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。请问年龄最小的员工生日是哪天?

9.6.2 最大/最小值输出

【例题9.40】某个部门中，员工的出生日期如下： 1974-11-20, 1995-04-11, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。请问年龄最小的员工生日是哪天?

解题思路

需要查找的是最小年龄对应的出生日期，但是年龄需要计算得到，可以用A.minp(x)计算。

	A	B
1	[1974-11-20, 1995-04-11, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13]	/生成生日序列
2	=A1.minp(age(~))	/求最小年龄者的生日
3	=A1.minp@a(age(~))	/求所有的最小年龄者的生日

A2	Value
	1995-04-11

A3	Index	Member
	1	1995-04-11
	2	1995-05-13

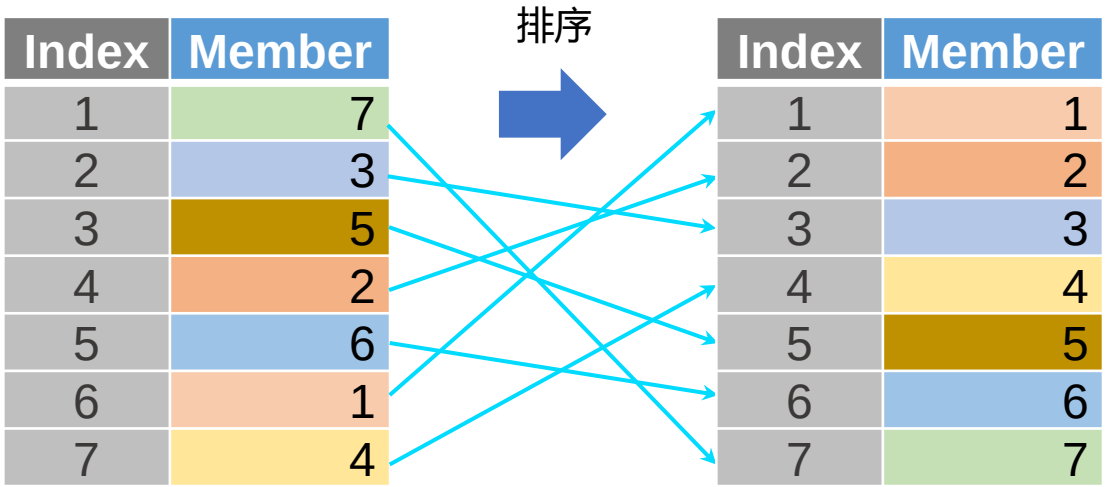
注意

- 查找最小年龄时，即使有多个同龄人，也只会返回第1位的生日，即使他可能不是所有人里面最小的那个；A3中添加**@a**选项，返回了“**所有**最小年龄者的生日”，此时会返回**序列**。

9.6.3 排序和排名

序列排序

- A.sort()函数将返回A升序排序后构成的新序列。
- 如果添加x参数，A.sort(x)仍然是用A中成员调整顺序构成新序列，但排序依据是根据A.(x) 。
- A.sort(x)函数可以添加选项@z， A.sort@z(x)将用降序来将A中成员重新排列成新序列。



9.6.3 排序和排名

【例题9.41】某次短跑比赛中，1~8跑道参赛的选手成绩依次如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。请按名次列出选手成绩?

9.6.3 排序和排名

【例题9.41】某次短跑比赛中，1~8跑道参赛的选手成绩依次如下（秒）：9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94。请按名次列出选手成绩?

解题思路

这里需要将数据排序，应该使用A.sort()函数。

	A	B
1	[9.98, 11.99, 9.80, 9.75, 9.79, 9.63, 9.88, 9.94]	/生成成绩序列
2	=A1.sort()	/各名次选手成绩

A2	Index	Member
	1	9.63
	2	9.75
	3	9.79
	4	9.80
	5	9.88
	6	9.94
	7	9.98
	8	11.99

注意

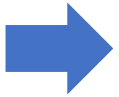
- A.sort()函数会生成新序列，而不会影响A本身。本例中，A2和A1中的结果是不同的。

前n名

- A.top(n)函数用来查找序列升序排序后，前n个成员构成的新序列。
- 如果添加x参数，A.top(n, x)的相当于A.(x).top(n)，先计算出A.(x)再返回它的前n名序号数列。注意这种情况是和前面的A.sort(x)不同的。
- A.top(n, x)函数并不支持@z选项，无法直接使用降序排序查找前n名，但是可以在n前添加负号实现降序排序的效果。

Index	Member
1	9.98
2	11.99
3	9.80
4	9.75
5	9.79
6	9.63
7	9.88
8	9.94

前3名



Index	前3名
1	9.63
2	9.75
3	9.79

【例题9.42】某个部门中，员工的出生日期如下： 1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。请问年龄最大的3位员工的生日?

9.6.3 排序和排名

前几名

【例题9.42】某个部门中，员工的出生日期如下： 1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13。请问年龄最大的3位员工的生日？

解题思路

需要查找“年龄最大的3位员工生日”，因此需要使用**降序**排序，需要**在排序表达式或n前加负号。**这里需要获得的仍然是员工的生日，所以应该用**A.top(n, x)**函数。这道题目，也可以转换为“生日最小的3位员工”来思考，用A.top(n)求解；但是由于年龄得到的是整数，两者并不完全一致，尤其是存在同龄员工的时候。

	A	B	C
1	[1974-11-20, 1980-07-19, 1970-12-17, 1985-03-07, 1984-07-07, 1995-05-13]		/生成生日序列
2	=A1.top(-3, age(~))	=A1.top(3, -age(~))	/年龄最大的3位员工生日

A2	Index	Member
	1	49
	2	45
	3	40

B2	Index	Member
	1	-49
	2	-45
	3	-40

注意

- 和**A.ptop(n, x)**的使用不太相同，使用**A.top(n, x)**函数时，返回的是A.(x)中的值构成的序列而不是序号数列。因此，这里B2中返回的年龄都变成了**负值**，这样的使用是**不对**的。
- **A.top(n, x)**中也可以添加选项**@1**，在查找**第1名或最后一名**时，返回第1名或最后1名的计算结果，此时对应的**n为1或-1。**

9.7 多维序列的计算

多维序列的计算

- 当序列的**每个成员都是序列**时，序列是**多维序列。**
- 多维序列**仍然是序列**，所以可以使用序列的各种函数。
- 序列的序列A中，可以用**A.pos@p(x)**查找某个序列成员**x**的位置，为了和普通序列的多成员查找做区别，需要添加**@p**选项。
- 序列的序列A中，可以用各个序列成员执行各种集合计算，包括**和列A.conj(), 并列A.union(), 差列A.diff(), 交列A.isect()**和**异或列A.xunion()**。

【例题9.43】已知3组学生某次测验的分数依次如下：1组：78，85，59；2组：66，82，96；3组：88，66，100, 69。求出这些学生的总平均分。

9.7 多维序列的计算

【例题9.43】已知3组学生某次测验的分数依次如下：1组：78，85，59；2组：66，82，96；3组：88，66，100, 69。求出这些学生的总平均分。

解题思路

这里的数据分成多组给出，实际上是个2维序列结构。可以分别用序列的循环函数，计算每组学生的总分的和得到总分数，计算每组学生人数的和得到总人数，再计算平均分。更简单的方法是先用所有序列成员计算合集，再调用序列函数计算总平均分。

	A	B	C
1	[[78,85,59], [66, 82, 96], [88, 66, 100, 69]]		/生成分数多维序列
2	=A1.sum(~.sum())	=A1.sum(~.count())	/计算总分和总人数
3	=A2/B2		/计算平均分，保留2位小数
4	=A1.conj().avg()		/先将所有学生成绩合并到一个序列，再计算平均分

A2	Value
	789

B2	Value
	10

A3	Value
	78.9

A4	Value
	78.9

9.7 多维序列的计算

【例题9.44】学校准备举办运动会，包括100m, 400m，3000m, Long jump, High jump, Javelin, Shot put等项目。下面是5位男同学的报名情况。1号：100m, Long jump； 2号：Shot put； 3号：100m, 400m, 3000m； 4号：Long jump, 100m； 5号：Shot put, 400m。请问哪位同学参加100m和Long jump这两项比赛？还有哪些项目没有人报名？

解题思路

这里的报名数据是个2维序列结构，查找所有报名某些项目同学的序号可以用A.pos@ap(p)，也可以用A.pselect@a(x)来查找。统计哪些项目无人报名，可以先在报名数据中计算并集，得到报名的所有项目，再用总项目序列和结果序列做差列；也可以把总项目序列添加到报表序列中，在2维序列中计算差列。

9.7 多维序列的计算

【例题9.44】学校准备举办运动会，包括100m, 400m，3000m, Long jump, High jump, Javelin, Shot put等项目。下面是5位男同学的报名情况。1号：100m, Long jump； 2号：Shot put； 3号：100m, 400m, 3000m； 4号：Long jump, 100m； 5号：Shot put, 400m。请问哪位同学参加100m和Long jump这两项比赛？还有哪些项目没有人报名？

	A	B	C
1	[100m, 400m，3000m, Long jump, High jump, Javelin, Shot put]	[[100m, Long jump], [Shot put], [100m, 400m, 3000m], [Long jump, 100m], [Shot put, 400m]]	/生成项目序列和报名情况的多维序列
2	=B1.pos@ap(["100m", "Long jump"])	=B1.pselect@a(~.contain("100m", "Long jump"))	/查找哪位同学报名了100米和跳远
3	=A1\B1.union()	=([A1]|B1).diff()	/计算哪些项目无人报名

A2	Index	Member
	1	1

B2	Index	Member
	1	1
	2	4

A3	Index	Member
	1	High jump
	2	Javelin

B3	Index	Member
	1	High jump
	2	Javelin

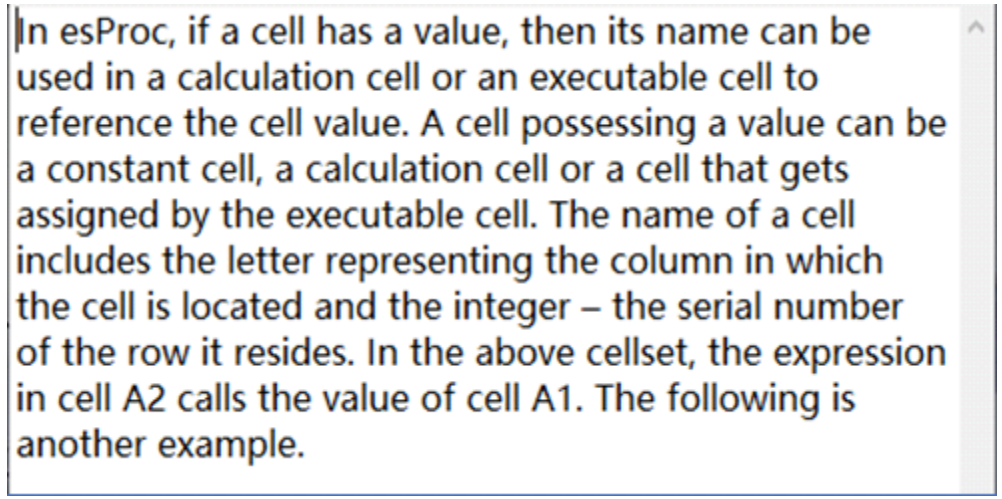
注意

- A2和B2中得到的结果有一些区别，这是由于1号和4号虽然都报名了100米和跳远，但是报名时顺序不同，所以直接用**A.pos@ap(p)**无法找到4号同学。在B2中用了**A.contain(xi,...)**作判断，忽略了序列中成员的顺序，同时也考虑了选手报名更多项目的情况。B2中的计算方法更为合理。另外需要注意，常数各种的字符串序列成员可以直接输入，表达式中的则需要添加引号。
- A3和B3中得到的结果是相同的，先计算并列再做差列，和合并后做差列的效果相同。

9.8 序列计算举例

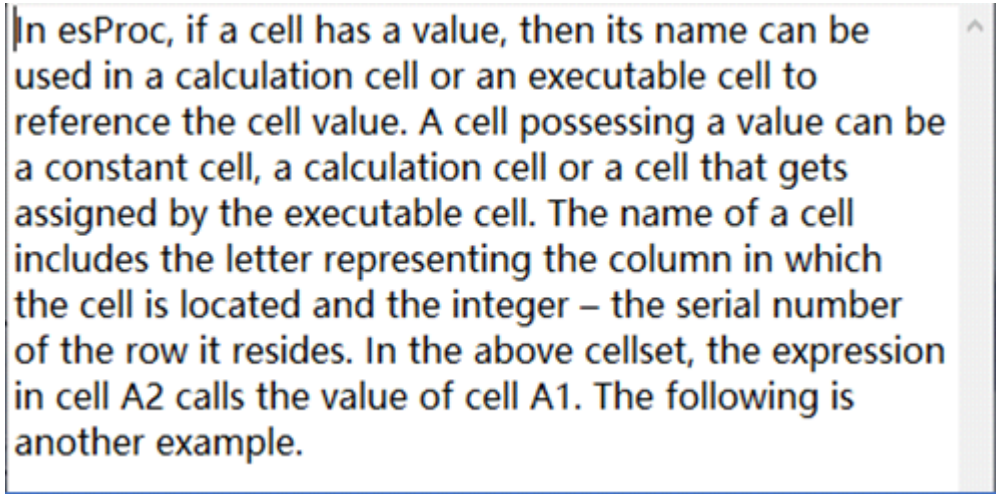
9.8 序列计算举例

【例9.45】读取一个文本，统计单词“the”的个数。



9.8 序列计算举例

【例9.45】读取一个文本，统计单词“the”的个数。



解题思路

读出文本生成字符串，再拆出单词序列。然后过滤出单词“the”组成的序列，最后统计成员的个数即可。

s.words()可以从字符串中拆出单词构成字符串序列。

A.select(x)可以选出序列中符合条件的成员。

A.len()可以取得序列中成员的个数。

	A	B
1	=file("text.txt").read().words()	/拆出单词构成序列
2	=A1.select(lower(~)=="the")	/筛选出"the"
3	>output("Count="/A2.len())	/统计个数

输出结果

Count=13

注意

- 本题中A2中选择出的单词不会有空值，所以用**A.len()**效率更高，不必使用A.count()计算，使用这个函数还需要排除非空成员。

9.8 序列计算举例

【例9.46】有一组七人裁判的跳水评分（7.8，8.3，8.1，8.5，8.0，8.1，8.0），评分规则：去掉一个最高分和一个最低分，合计出总分后，再乘以动作难度系数（2.7），最后再乘以3/5。

9.8 序列计算举例

【例9.46】有一组七人裁判的跳水评分（7.8，8.3，8.1，8.5，8.0，8.1，8.0），评分规则：去掉一个最高分和一个最低分，合计出总分后，再乘以动作难度系数（2.7），最后再乘以3/5。

解题思路

此题按评分规则，调用SPL相关专用函数进行计算即可。

A.delete(k)可以删除序列A中指定的成员。

A.pmax()和**A.pmin()**可以获取序列中最大/最小成员的位置序号。

A.sum()可以算出序列中所有非空成员的汇总值。

	A	B
1	[7.8,8.3,8.1,8.5,8.0,8.1,8.0]	/七人裁判的跳水评分
2	2.7	/该选手动作难度系数
3	>A1.delete(A1.pmax())	/去掉一个最高分
4	>A1.delete(A1.pmin())	/去掉一个最低分
5	=A1.sum()	/合计算出总分
6	=A5*A2*3/5	/算出选手最终得分

A6	Value
	65.61

9.8 序列计算举例

【例9.47】某8人学习小组参加数学模拟考试，成绩：89.5，69.5，78，66，45.5，72.5，43.5，59.5。其中小强考了66分，想知道他在小组中的名次以及该小组的平均分。

9.8 序列计算举例

【例9.47】某8人学习小组参加数学模拟考试，成绩：89.5，69.5，78，66，45.5，72.5，43.5，59.5。其中小强考了66分，想知道他在小组中的名次以及该小组的平均分。

解题思路

排序后定位小强考试分数在小组中的名次，再求出平均分对比即可。

A.sort()可以对序列成员进行排序。

A.pos(x)可以获取序列成员的位置序号。

A.avg()可以对序列中非空成员求平均值。

	A	B
1	[89.5,69.5,78,66,45.5,72.5,43.5,59.5]	/该学习小组全部数学成绩
2	66	/小强的成绩
3	=A1.sort@z()	/小组中成绩按由高到低作排序
4	=A3.pos(66)	/定位小强在小组中的排名
5	=A1.avg()	/算出小组的平均成绩

A4	Value
	5

即小强成绩小组中排名第5

A5	Value
	65.5

小组平均分为65.5

9.8 序列计算举例

【例题9.48】不调用SPL的特殊函数，使用矩阵法解二元一次方程$\begin{cases} 2*x+y=5 \\ x+2*y=4 \end{cases}$

9.8 序列计算举例

【例题9.48】不调用SPL的特殊函数，使用矩阵法解二元一次方程$\begin{cases}2*x+y=5\\x+2*y=4\end{cases}$

解题思路

创建方程的增广矩阵，用加减消元法，将y的系数化为0后再将对应的x的系数化为1，即可求得x的值。再将求得的x值代入y的系数不为0的另一个等式中，将y的系数化为1后即可求得y的值。

	A	B
1	[[2,1,5],[1,2,4]]	/创建方程组的增广矩阵
2	>A1(1)=A1(1).(~*A1(2)(2))--A1(2).(~*A1(1)(2))	/加减消元，消去第1个方程中的y
3	>A1(1)=A1(1).(~/A1(1)(1))	/在第1个方程中求出x值
4	>A1(2)=A1(2)--A1(1).(~*A1(2)(1))	/加减消元，在第2个方程中消去x
5	>A1(2)=A1(2).(~/A1(2)(2))	/求出y值
6	>output("x="/A1(1)(3)/",y="/A1(2)(3))	/输出结果

输出结果：x=2.0, y=1.0

即方程的解为：$\begin{cases}x=2\\y=1\end{cases}$

9.9 习题

9.9 习题

【习题9.1】已知30以内的所有质数，怎样快速求出30以内的所有合数?

【习题9.2】假设某人坐火车去某地办公，固定的行程安排为：第一周的周四买票过去，则第二周的周二买票回来，然后第三周的周四再买票过去……如此循环往复。已知：2020年3月5号（周四），预定要过去某地办公。现想知道：从3月初到4月底，具体需要买哪些日子的去程票和返程票。

【习题9.3】读取文本中保存的订单金额，统计其中金额大于等于1000的订单的数量。

【习题9.4】分别用递推法和通项公式法，构造包含50项成员的斐波那契数列，并验证二者是完全相等的。其中通项公式：$a_n = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n]$

【习题9.5】有10个人报名，通过考试成绩（59，63，52.5，57，48，63.5，66，32.5，45，55）录取前6名，分数线应该怎么划?

【习题9.6】有文件名为"filename@www.xxxx.com.rar"形式的字符串，想获得将其中的"@www.xxxx.com"去除后的文件名字符串。

【习题9.7】利用序列求无限循环小数的循环节，以 47/19 为例。

【习题9.8】试用滑动平均滤波算法（N=4），对一组采集的电池电压数据（36.4，36.5，36.8，36.3，36.5，36.6，36.3，35.3，36.8，36.4，36.1，36.5，36.2，36.7，36.5，36.3）进行数字滤波处理。

9.9 习题

【习题9.1】已知30以内的所有质数，怎样快速求出30以内的所有合数?

解题思路

正整数包括质数、合数与1。从30以内正整数的集合中去除30以内的质数集合再去除1，即为30以内的合数集合。我们可以用序列的差集运算来处理此题。

	A	B
1	[2,3,5,7,11,13,17,19,23,29]	/1~30以内质数的集合
2	=to(30)\A1\1	/用差集运算求合数集合

A2	Index	Member
	1	4
	2	6
	3	8
	4	9
	5	10
	6	12
	…	…

9.9 习题

【习题9.2】假设某人坐火车去某地办公，固定的行程安排为：第一周的周四买票过去，则第二周的周二买票回来，然后第三周的周四再买票过去……如此循环往复。已知：2020年3月5号（周四），预定要过去某地办公。现想知道：从3月初到4月底，具体需要买哪些日子的去程票和返程票。

解题思路

通过简单推算可知：去程日期之间相隔14天，回程日期同理。最近的去程日期为3月5号，最近的回程日期为3月10日。

SPL中**periods()**函数可以按照指定时间间隔获取日期时间序列，比如使用**@y**为以年为单位，**无选项**为以日为单位，使用**@s**为以秒为单位等。

	A	B	C
1	2020-03-01	2020-04-30	
2	=periods(A1,B1,1)		/将日期分割成序列
3	=A2.step(14,5)		/A3为全部去程日期
4	=A2.step(14,10)		/A4为全部回程日期

去程

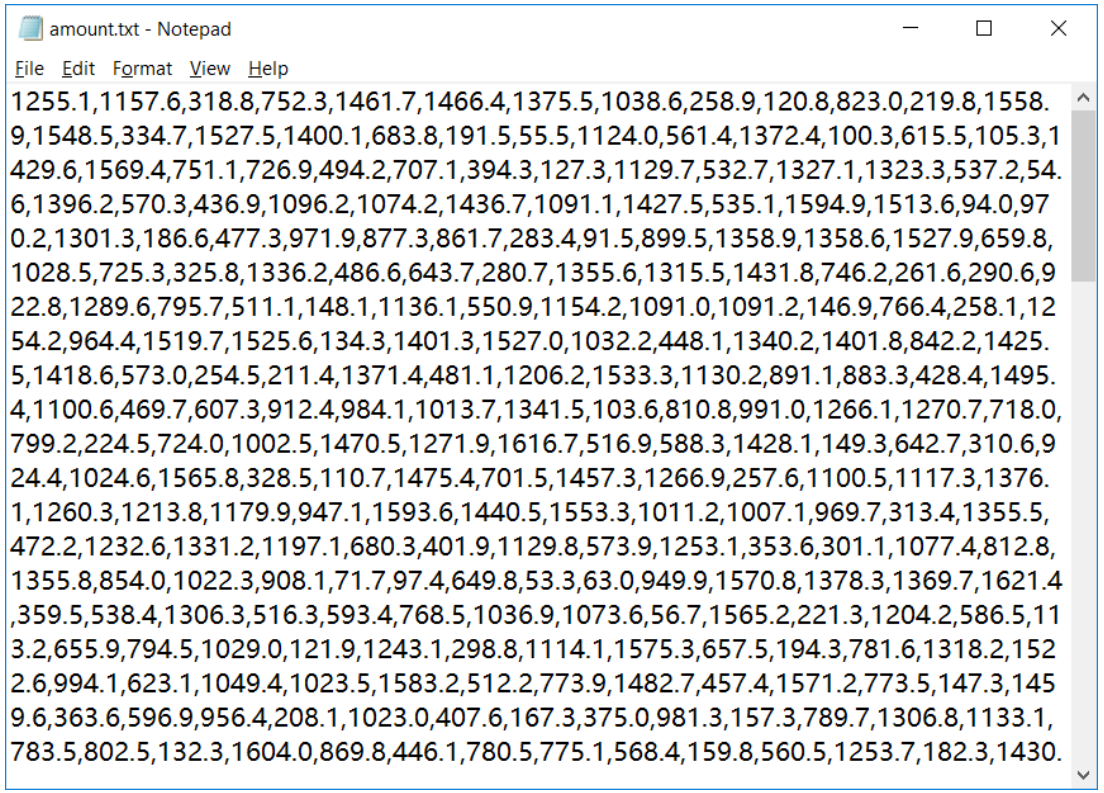
A3	Index	Member
	1	2020-03-05
	2	2020-03-19
	3	2020-04-02
	4	2020-04-16
	5	2020-04-30

回程

A4	Index	Member
	1	2020-03-10
	2	2020-03-24
	3	2020-04-07
	4	2020-04-21

9.9 习题

【习题9.3】读取文本中保存的订单金额，统计其中金额大于等于1000的订单的数量。



解题思路

先筛选出订单金额大于等于1000的数据，再统计数量即可。

	A	B
1	=file("amount.txt").read().split@cp()	/@p:文本数据自动解析类型
2	=A1.select(~>=1000)	/只保留大于等于1000的数据
3	=A2.count()	/统计总数

A3	Value
	419

9.9 习题

【习题9.4】分别用递推法和通项公式法，构造包含50项成员的斐波那契数列，并验证二者是完全相等的。

其中通项公式：$a_n = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n]$

解题思路

用序列循环函数，用两种方法生成斐波那契数列并求两个序列的差集，差集为空列则证明完全相等。

	A	B	C
1	50	[1,1]	
2	>A1.run(if(#>2, B1.insert(0, B1.m(-1)+B1.m(-2))))		/递推法在B1中生成斐波那契数列
3	=A1.(long(round(1/sqrt(5)*(power((1+sqrt(5))/2,~)-power((1-sqrt(5))/2,~)))))		/公式法生成斐波那契数列
4	=[B1,A3].diff()		/空列证明无差别

B1	Index	Member
	1	1
	2	1
	3	2
	4	3
	5	5
	6	8
	…	…

A3	Index	Member
	1	1
	2	1
	3	2
	4	3
	5	5
	6	8
	…	…

A4	Index	Member

9.9 习题

【习题9.5】有10个人报名，通过考试成绩（59，63，52.5，57，48，63.5，66，32.5，45，55）录取前6名，第6名的分数是多少?

解题思路

可以先将考试成绩由高到低做一次排序，再取前6名，则第6名的成绩即为分数线。

SPL中**A.top()**函数适用于在原始数据中取出前N名的操作，比执行A.sort()后再执行A.to()的效率更高

	A	B
1	[59,63,52.5,57,48,63.5,66,32.5,45,55]	/考试成绩序列
2	=A1.top(-6;~)	/按降序排序取前6名
3	=A2(6)	/以第6名的成绩划线

A3	Value
	55

9.9 习题

【习题9.6】有文件名为“filename@www.xxxx.com.rar”形式的字符串，想获得将其中的"@www.xxxx.com"去除后的文件名字符串。

解题思路

用**s.split(d)**函数将文件名分割为字符串序列，将其中的一部分作删除处理后，剩下的拼接返回。

	A	B
1	filename@www.xxxx.com.rar	/将字符串分割为字符序列
2	=A1.split()	/将字符串分割为字符序列
3	=A2.delete(to(A2.pos@z("@"),A2.pos@z(".")-1)).concat()	/删除不需要的子串后再将字符序列合并回字符串

A3	Value
	filename.rar

注意

- 也可以使用字符串函数实现同样功能：=left(A1,pos@z(A1, "@")-1)或=right(A1,len(A1)-pos@z(A1, ".")+1)

9.9 习题

【习题9.7】利用序列求无限循环小数的循环节，以 47/19 为例。

解题思路

可以使用比较余数的方法来确定循环小数的循环节。

	A	B	C	D
1	47	19		/A1为被除数，B1为除数
2	=A1%B1			/A2做取余计算得到第一次循环要使用的余数
3	for	=B5.pos(A2)		/A3定义无限循环，B3确定本次循环使用余数出现的位置
4		if B3 > 0	break	/如果本次循环使用余数在之前循环出现过则结束本次循环
5		=B5 | A2		/B5保存每次计算的余数
6		=B6 | int(A2*10/B1)		/B6保存每次计算的商，这就是循环节部分
7		>A2 = A2*10%B1		/A2更新为下次循环要使用的余数
8	=B6.to(B3,).concat()			/B6中从位置B3开始到最后即为循环节，取出拼成字符串

A8	Value
	4736842105263157 89

9.9 习题

【习题9.8】试用滑动平均滤波算法（N=4），对一组采集的电池电压数据（36.4，36.5，36.8，36.3，36.5，36.6，36.3，35.3，36.8，36.4，36.1，36.5，36.2，36.7，36.5，36.3）进行数字滤波处理。

解题思路

所谓滑动平均滤波，即在数据缓冲区中，依顺序放入N个采样数据（此题取N=4），并算出算术平均值。然后每纳入一个新数据，就将最早采样的那个数据丢弃，再求出包括新数据在内的N个数据的算术平均值，以此循环往复，直至处理完所有的数据。在序列循环中，可以用相对引用区间~[a, b]的方式获取采样数据计算平均值。

	A	B	C
1	[36.4, 36.5, 36.8, 36.3, 36.5, 36.6, 36.3, 35.3, 36.8, 36.4, 36.1, 36.5, 36.2, 36.7, 36.5, 36.3]	[]	/输入采集数据，用B1准备记录滤波结果
2	>A1.run(if(#>3, B1=B1|round(~[-3,0].avg(), 2)))		/循环采集数据，从第4个起相对引用4个数据计算均值，加入滤波结果序列

B1	Index	Member
	1	36.5
	2	36.53
	3	36.55
	4	36.42
	5	36.17
	6	36.25
	7	36.2
	8	36.15
	9	36.45
	10	36.3
	11	36.38
	12	36.48
	13	36.42

第10章 数据表

10.1 结构化数据

- 结构化数据
- 字段与记录
- 数据表

10.2 序表的操作

- 序表创建/查看
- 记录访问
- 基本增删改
- 举例与延伸操作

10.3 外部数据

- 数据文件
- 文件数据读取
- 文件数据写出
- 数据库查询

10.4 序表的运算

- 聚合
- 过滤

10.5 排列

- 排列概念
- 排列的运算
- 排列与序表的不同

10.6 习题

10.1 结构化数据

10.1.1 结构化数据

我们处理数据时会遇到下面类似结构的数据。

学生成绩表：

	A	B	C	D	E	F
1	Index	Name	Chinese	Math	English	TotalScore
2	70605	YangLu	131	143	144	418
3	70603	WangXue	131	135	144	410
4	70609	HanLin	127	139	142	408
5	70601	ShaLongYu	123	148	136	407
6	70606	LiJianXue	126	135	140	401
7	70604	HanYuPeng	129	133	138	400

如YangLu的成绩信息是一行，每行包括考号、姓名、语文、数学等信息。

员工出勤表：

Date	Sid	Name	StartTime	EndTime	Attendance
2020-03-01	1012	Tom	08.59	18.30	Normal
2020-03-01	1015	Jack	08.52	18.00	Normal
2020-03-01	1016	Adam	08.43	18.05	Normal
…	…	…			

如员工Tom的考勤信息是一行，每行包括日期、员工号、员工名称等信息。

订单表

No	Name	Price	Num	Details
20200501	banana	5.00	5	
20200502	apple	7.50	7	From Chengdu
20200503	cucumber	4.60	5	
20200504	orange	6.50	2	
20200505	egg	8.60	5	

如banana订单信息是一行，每行包括序号、商品名称、价格等信息。

单位来访人员登记表

No	Name	Gender	IDCard	Tel	PDate	PTime	Content	Comment
1	Hando	M	xxx	123456789	2020-05-02	10:25	Client Review	

如Hando来访人员的登记信息是一行，每行包括序号、姓名、性别等信息。

股票行情表

No	Symbol	Name	LastTrade	Chg	Change
1	688123	Juchen Co., Ltd	73.30	15.23%	9.69
2	688037	Core micro	118.88	12.19%	12.92
3	002347	Tel shares	4.57	10.12%	0.42
4	601616	SGEG	3.81	10.12%	0.35
5	300165	SKYRAY	4.90	10.11%	0.45

如SGEG股票的行情信息是一行，每行包括序号、股票代码、名称、最新价等信息。

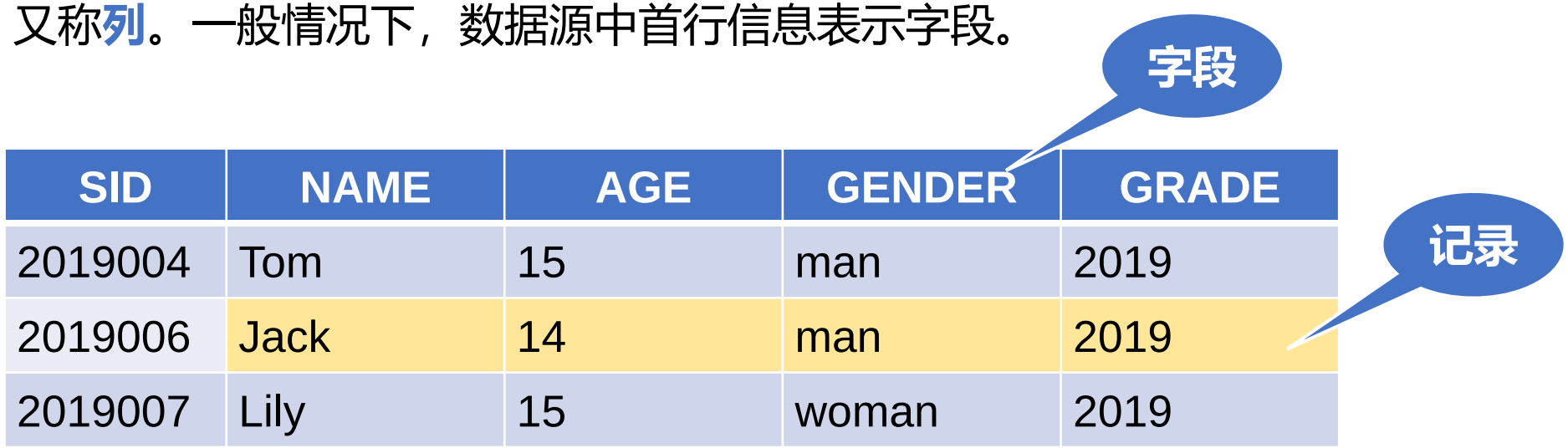
10.1.1 结构化数据

从上面的例子看到，虽然内容不一样、但它们都由一行行信息构成，每行信息由若干不同的属性组成，类似这样的数据就是**结构化数据。**

Index	CLASS	STUDENTID	SUBJECT	SCORE
1	Class one	1	English	84
2	Class one	1	Math	77
3	Class one	1	PE	69
4	Class one	2	English	81
5	Class one	2	Math	80

10.1.2 字段与记录

学生登记表属于结构化数据，要求填写学号、姓名、年龄、性别等属性信息，我们称这些属性为**字段**, 又称**列**。一般情况下，数据源中首行信息表示字段。



SID	NAME	AGE	GENDER	GRADE
2019004	Tom	15	man	2019
2019006	Jack	14	man	2019
2019007	Lily	15	woman	2019

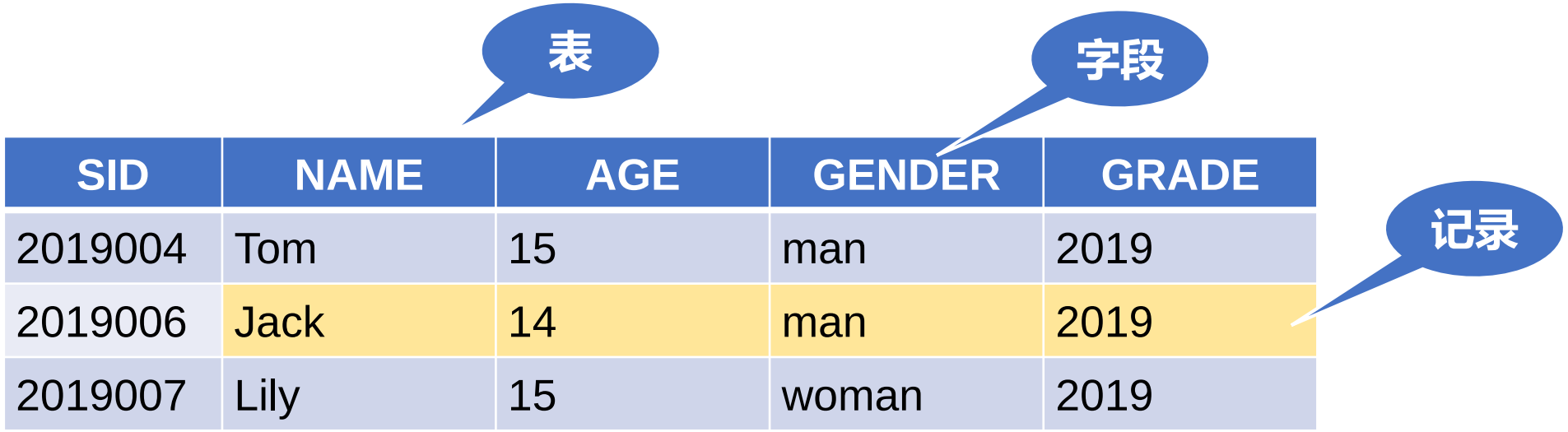
登记表中每一行都表示数据**记录**，一条记录由若干个字段值组成。

即**行**表示记录，**列**表示字段，每一行表示一条记录，每一列表示一个字段

有记录有字段的这批表格数据，我们称为**数据表**

10.1.3 数据表

数据表的字段集合称为**数据结构**，简称结构。在学生登记表中，每个学生信息包含的 SID, NAME、AGE、GENDER、GRADE这些字段的集合就是这个数据表的数据结构。
有时候我们也称数据结构是属于记录的，因为每一条记录也都有这些字段。

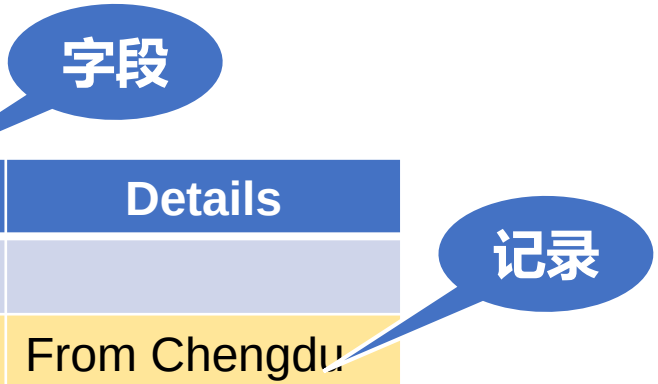


SID	NAME	AGE	GENDER	GRADE
2019004	Tom	15	man	2019
2019006	Jack	14	man	2019
2019007	Lily	15	woman	2019

数据表简称为**表**，由一批**结构相同**的数据（**记录**）构成。
数据表也常常称为二维表，因为表现出来是个行列二维结构，表中的数据可以用其所在单元格位置的行号、列号来确定，以方便读写信息。如Lily所在位置是第3行第2列。

10.1.3 数据表

我们再看看前面所提到的订单表。



No	Name	Price	Num	Details
20200501	banana	5.00	5	
20200502	apple	7.50	7	From Chengdu
20200503	cucumber	4.60	5	
20200504	orange	6.50	2	
20200505	egg	8.60	5	

数据表**字段**有No、Name、 Price、Num、Details。

记录表示一条购物信息，如第2行记录的字段值分别为**20200502、apple、7.50、7 、From Chengdu**等。

通常用行、列表示访问数据在表中的位置。如**apple**在第2行第2列。

10.1.3 数据表

股票行情表中的字段、记录

No	Symbol	Name	LastTrade	Chg	Change
1	688123	Juchen Co., Ltd	73.30	15.23%	9.69
2	688037	Core micro	118.88	12.19%	12.92
3	002347	Tel shares	4.57	10.12%	0.42
4	601616	SGEG	3.81	10.12%	0.35
5	300165	SKYRAY	4.90	10.11%	0.45

字段有No、Symbol 、Name、 LastTrade、Chg、Change等。
记录表示一只股票行情信息，如第2行记录的字段值分别为688037、Core micro、118.88等。
要访问数据可用行、列描述。如Tel shares在第3行第3列。

10.2 序表的操作

10.2.1 序表创建/查看

在建立数据表前，先要确定数据结构是什么，也就是都有什么字段。

比如用户信息表，它由序号、用户名、姓别、年龄4个字段。有了结构信息，就可以建表了。

SPL中的数据表也叫序表，后面会常用序表来表示数据表。

	A	B
1	=create(SID,NAME, GENDER, AGE)	/创建用户表

A1返回以SID, NAME, GENDER, AGE四个字段为结构的空表，它只有结构还没有记录。

Index	SID	NAME	GENDER	AGE

显示时，表格中的Index列只是辅助列，表示序表中的记录序号，不是序表中的真实字段，这和序列中用来显示成员序号中的Index列作用是相同的。

10.2.1 序表创建/查看

有了用户信息表之后， 就可以用T.record(A)将序列A数据依次写入表T中。

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME, GENDER, AGE)	>A4.record([A1:D3])	/record用于初始填数	

[A1:D3]为准备的数据，执行脚本后，A4表数据如下：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11

T.record(A)： T序表中用A的成员依次追加记录的每个字段值。

10.2.2 记录访问

序表可以理解为是**记录构成的序列**，可以用**T(k)**访问其成员，其中k为记录在表中的行位置。

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME,GENDER, AGE)	>A4.record([A1:D3])		
5	=A4			/访问序表
6	=A4(1)			/获取表中第1条记录
7	=A4(3)			/获取表中第3条记录

A5	SID	NAME	GENDER	AGE
	1	Jack	M	12
	2	Tom	M	15
	3	July	F	11

A6	SID	NAME	GENDER	AGE
	1	Jack	M	12

A7	SID	NAME	GENDER	AGE
	3	July	F	11

10.2.2 记录访问

字段被认为是记录的属性，在计算机语言中，一般使用.操作符，即r.F的语法来访问访问字段值；有时也可以使用序号r.#i的方式访问，其中#i为第i个字段。

	A	B	C	D
1	1	Tom	M	15
2	2	July	F	11
3	=create(SID,NAME, GENDER, AGE)	>A3.record([A1:D2])		/序表
4	=A3(1)			/获取首条记录
5	=A4.NAME			/获取记录中的字段NAME的值
6	=A4.#2			/获取记录中第2个字段(NAME)的值
7	=A3(1).NAME			/获取第1条记录中的字段NAME的值

A3

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	11

记录

A4

SID	NAME	GENDER	AGE
1	Tom	M	15

字段值

A5

Value
Tom

A6、A7中的结果和A5是相同的

10.2.2 记录访问

访问第N条记录的某字段值，如访问第3条记录的用户名July等。

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME,GENDER,AGE)	=A3.record([A1:D2])		/序表
5	=A4(3).NAME			/访问第3条记录的用户名
6	=A4(3).#2			/同上
7	=A4(1).AGE			/访问第1条记录的用户年龄
8	=A4(1).#4			/同上
9	=A4(1).array()			/获取第1条记录所有字段的值组成的序列
10	=A4(1).array(NAME, AGE)			/获取第1条记录NAME和AGE两个字段组成的序列

10.2.2 记录访问

对记录的字段值进行修改。 r.F=x修改记录r的F字段值为x，也可用r.#i=x方式，其中#i表示记录r的第i个字段。

	A	B	C	D
1	1	Tom	M	15
2	2	July	F	11
3	=create(SID,NAME, GENDER, AGE)	=A3.record([A1:D2])		/序表
4	=A3(1)			/访问首条记录，返回记录A4
5	>A4.NAME="Jack"			/修改记录字段NAME的值
6	>A4.#2="Jack"			/修改记录字段NAME的值

A4

SID	NAME	GENDER	AGE
1	Tom	M	15

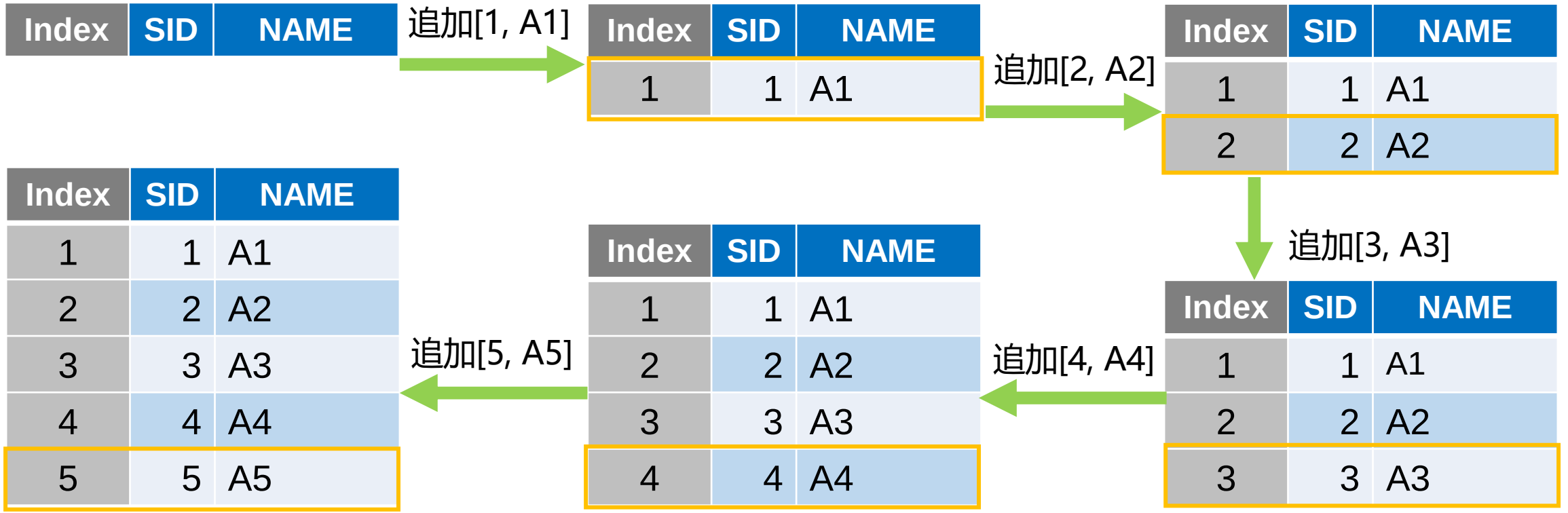
修改

A4

SID	NAME	GENDER	AGE
1	Jack	M	15

10.2.3 基本增删改

追加记录：在序表最后增加新记录



新追加的数据放在表的末尾位置，数据按追加顺序依次排列。

10.2.3 基本增删改

追加记录，用**T.insert(0,x_i,…)**在序表**T**中**追加**新创建的记录。

	A	B
1	=create(SID,NAME, GENDER, AGE)	/建立空表A1
2	>A1.insert(0, 1,"Tom","M",15)	/添加1条记录
3	>A1.insert(0, 2,"July","F", 11)	/追加1条记录
4	>A1.insert(0, 3,"Gail","M", 13)	/追加1条记录

Index	SID	NAME	GENDER	AGE

A2追加

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15

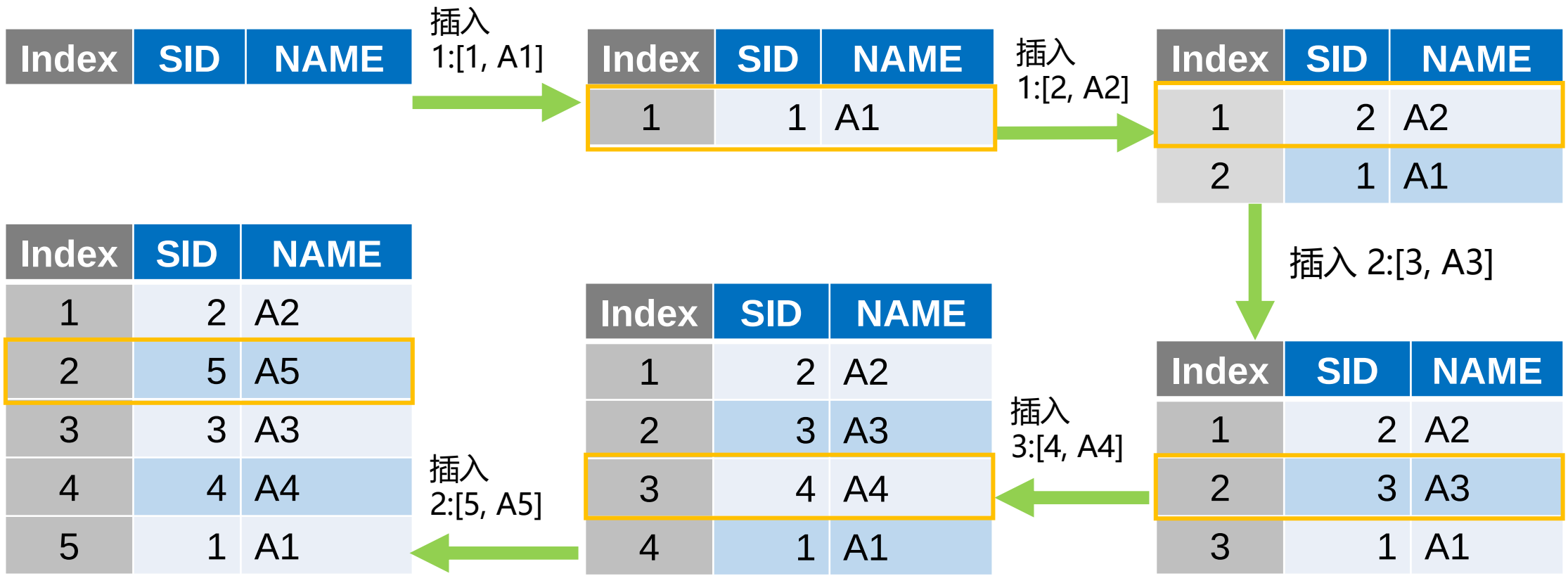
Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	11

A4追加

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	11
3	3	Gail	M	13

10.2.3 基本增删改

插入记录：在序表指定位置插入新创建的记录，此位置后的记录相应地下移



在指定位置插入记录，新增记录存放在指定位置，指定位置后的记录依次向下移动。

10.2.3 基本增删改

插入记录，T.insert(k,...)，在表中原第k行之前插入新创建的记录

	A	B
1	=create(SID,NAME, GENDER, AGE)	/建立空表A1
2	>A1.insert(1, 1,"Tom","M",15)	/添加1条记录
3	>A1.insert(2, 2,"July","F", 11)	/记录插入第2行
4	>A1.insert(2, 3,"Gail","M", 13)	/记录插入第2行

Index	SID	NAME	GENDER	AGE

A2插入

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15

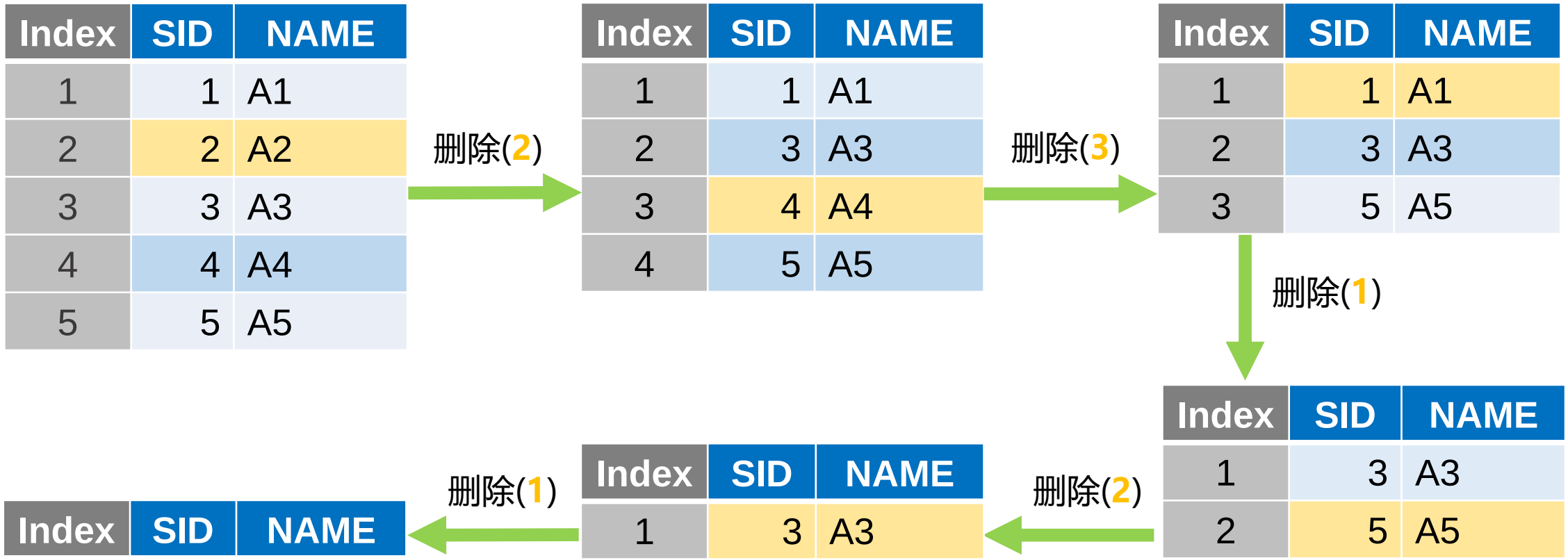
Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	11

A4插入

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	3	Gail	M	13
3	2	July	F	11

10.2.3 基本增删改

删除记录：删除指定位置的记录，后面记录将上移



注意删除记录过程中，记录行号的变化。

10.2.3 基本增删改

删除记录，T.delete(k)，删除第k行记录

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11
5	=create(SID,NAME, GENDER, AGE)	=A5.record([A1:D4])		/序表
6	>A5.delete(2)	>A5.delete(1)		/A6删除表中第2行记录，B6删除第1行记录
7	>A5.delete(1)			/A7删除第1行记录

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

删除(2)

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	3	July	F	11
3	4	Gail	M	11

删除(1)

Index	SID	NAME	GENDER	AGE
1	3	July	F	11
2	4	Gail	M	11

删除(1)

Index	SID	NAME	GENDER	AGE
1	4	Gail	M	11

10.2.3 基本增删改

修改记录：修改指定行号的记录的字段值

Index	SID	NAME
1	1	A1
2	2	A2

修改1:[11]

Index	SID	NAME
1	11	A1
2	3	A3

修改第1行第1个字段值

Index	SID	NAME
1	1	A1
2	2	A2

修改1:[A12:NAME]

Index	SID	NAME
1	1	A12
2	3	A3

修改第1行NAME字段值

Index	SID	NAME
1	1	A1
2	2	A2

修改2:[30，A30]

Index	SID	NAME
1	1	A1
2	30	A30

修改第2行第1、2字段值

10.2.3 基本增删改

修改记录，T.modify (k, x)修改第k行记录的字段

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME, GENDER, AGE)	=A4.record([A1:D3])		/数据填入表中。
5	>A4.modify(2,22)			/修改第2行第1列字段值
6	>A4.modify(3,"July33":NAME, 9:AGE)			/修改第3行指定字段的值
7	>A4.modify(1,"Jack11":NAME ,"Male")			/数据无字段名时按已有字段次序修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11

A5修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	22	Tom	M	15
3	3	July	F	11

A6修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	22	Tom	M	15
3	3	July33	F	9

A7修改

Index	SID	NAME	GENDER	AGE
1	1	Jack11	Male	12
2	22	Tom	M	15
3	3	July33	F	9

10.2.3 基本增删改

修改记录：直接修改指定记录的字段值

SID	NAME	GENDER	AGE
1	A1	M	10

修改(11)

SID	NAME	GENDER	AGE
11	A1	M	10

修改第1个字段值

SID	NAME	GENDER	AGE
1	A1	M	10

修改(A12:NAME)

SID	NAME	GENDER	AGE
1	A12	M	10

修改指定NAME字段值

SID	NAME	GENDER	AGE
1	A1	M	10

修改(A12:NAME, 15:AGE)

SID	NAME	GENDER	AGE
1	A12	M	15

修改指定多个字段值

SID	NAME	GENDER	AGE
1	A1	M	10

修改(30, A30, Male)

SID	NAME	GENDER	AGE
30	A30	Male	10

数据无字段名时按字段次序修改

SID	NAME	GENDER	AGE
1	A1	M	10

修改(A12:NAME, Male, 15)

SID	NAME	GENDER	AGE
1	A12	Male	15

数据无字段名时按已有字段次序修改

10.2.3 基本增删改

修改记录，r.modify(x)修改记录r的字段

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME, GENDER, AGE)	>A4.record([A1:D3])		/表中填入数据
5	=A4(2)	>A5.modify(22)		/修改记录的首个字段值
6	>A5.modify("Tom22":NAME)			/修改记录的指定字段NAME值
7	>A5.modify(44,"Tom44","Male")			/数据无字段名时按字段次序修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11

A5修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	22	Tom	M	15
3	3	July	F	11

A6修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	22	Tom22	M	15
3	3	July	F	11

A7修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	44	Tom44	Male	15
3	3	July	F	11

10.2.3 基本增删改

修改记录，也可以使用r.F=a 或 r.#i=a 修改记录

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME, GENDER, AGE)	>A4.record([A1:D3])		/序表
5	>A4(2).#1 = 22			/修改第2行第1列字段值
6	>A4(3).NAME="July33"			/修改第3行NAME字段值
7	>A4(3).GENDER="Female"			/修改第3行GENDER字段值

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11

A5修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	22	Tom	M	15
3	3	July	F	11

A6修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	22	Tom	M	15
3	3	July33	F	11

A7修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	22	Tom	M	15
3	3	July33	Female	11

10.2.4 举例与延伸操作

【例10.1】序表除了直接用create()创建外，也可通过已有的表T.create()或记录r.create()创建同结构的表。

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME, GENDER, AGE)	>A4.record([A1:D3])		/填写数据
5	=A4.create()			/建立与A4同结构的空序表
6	=A4(1).create()			/建立与A4(1)同结构的空序表
7	=A6.record([A1:D3])			/填写数据

A4	SID	NAME	GENDER	AGE
	10	Jack	M	12
	11	Tom	M	15
	12	July	F	11

A5	SID	NAME	GENDER	AGE

A6	SID	NAME	GENDER	AGE
	10	Jack	M	12
	11	Tom	M	15
	12	July	F	11

10.2.4 举例与延伸操作

【例10.2】通过记录、表的行、列访问或字段名访问

	A	B	C	D
1	1	Jack	M	2007-05-03
2	2	Tom	M	2004-07-12
3	3	July	F	2010-09-04
4	4	Gail	M	2010-03-08
5	=create(SID,NAME,GENDER, BIRTHDAY)	>A5.record([A1:D4])		/初始化序表
6	=A5(2)			/访问第2条记录
7	=A6.SID	=A6.#1		/获取记录中字段SID的值
8	=A6.NAME	=A6.#2		/获取记录中字段NAME的值
9	=A5(2).NAME	=A5(2).#2		/获取第2条记录中字段NAME的值
10	=A5(4).BIRTHDAY	=A5(4).#4		/获取第4条记录中字段BIRTHDAY的值

10.2.4 举例与延伸操作

【例10.3】 增加记录出现空记录现象。T.insert(k,x)插入时指定的行k比表中记录数多2及以上时会补入字段值为空的记录

	A	B
1	=create(SID,NAME, GENDER, AGE)	/建立空表A1
2	>A1.insert(1, 1,"Tom","M",15)	/添加1条记录
3	>A1.insert(2, 2,"July","F", 11)	/再添加1条记录
4	>A1.insert(4, 3,"Gail","M", 13)	/再添加1条记录

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15

A3插入

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	11

A4插入

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	11
3	(NULL)	(NULL)	(NULL)	(NULL)
4	3	Gail	M	13

A4中k=4比表记录条数2多2，插入时就会出现第3行为空记录。

10.2.4 举例与延伸操作

【例10.4】将记录数据依次填充到表T的新记录里。T.insert@r(k:[r])

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME, GENDER, AGE)	>A4.record([A1:D3])		/序表
5	=A4.create()			/建立与A4同结构的空表A5
6	=[A4(1), A4(3)]			/获取A4表中第1、3行记录
7	>A5.insert@r(2: A6)			/从第2个位置开始，将记录插入到表A5中
8	>A5.insert@r(0: [A4(2)])			/将A4表中第2条记录追加到A5中

A4

SID	NAME	GENDER	AGE
1	Jack	M	12
2	Tom	M	15
3	July	F	11

A5

SID	NAME	GENDER	AGE
(null)	(null)	(null)	(null)
1	Jack	M	12
3	July	F	11
2	Tom	M	15

10.2.4 举例与延伸操作

【例10.5】表T中插入按字段名匹配写入记录，可用T.insert@f(k:[r]) 。

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME,GENDER,AGE)	>A4.record([A1:D3])		
5	=create(NAME,SID,AGE,GENDER)	=create(NAME, AGE)		/A5建立同字段名不同字段顺序的空表
6	=A4(2)			/读取第2条记录
7	>A5.insert@f(1: [A6])	>B5.insert@f(1: [A6])		/将记录插入到序表中

A4	SID	NAME	GENDER	AGE
	1	Jack	M	12
	2	Tom	M	15
	3	July	F	11

A5	NAME	SID	AGE	GENDER
	Tom	2	15	M

B5	NAME	AGE
	Tom	15

插入与表有相同字段名的记录时，则按相同字段名对应的字段值写入，不匹配的字段则忽略。

10.2.4 举例与延伸操作

【例10.6】用r.modify(x)修改记录r的字段值。

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME, GENDER, AGE)	>A4.record([A1:D3])	=A4(3)	/访问第3条记录
5	>C4.modify("July3":NAME,9:AGE)			/修改记录指定字段值
6	>C4.modify(33,"July33" ,"Female")			/数据无字段名时按字段次序修改
7	>C4.modify("July34":NAME,"F", 11)			/数据无字段名时按已有字段次序修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11

A5修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July3	F	9

A6修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	33	July34	F	11

A7修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	33	July33	Female	9

10.2.4 举例与延伸操作

【例10.7】用T(k).modify(x)修改第k行记录的字段值，等同于T.modify(k, x)。

	A	B	C	D
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	=create(SID,NAME, GENDER, AGE)	>A4.record([A1:D3])		/序表
5	>A4(3).modify("July3":NAME,9:AGE)			/修改第3条记录指定字段值
6	>A4(3).modify(33,"July33" ,"Female")			/数据无字段名时按字段次序修改
7	>A4(3).modify("July34":NAME, "F", 11)			/数据无字段名时按已有字段次序修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11

A5修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July3	F	9

A7修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	33	July33	Female	9

A7修改

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	33	July34	F	11

10.3 外部数据

10.3.1 数据文件

在计算机日常使用中，经常遇到csv，txt，xls等数据文件格式。

txt格式：常用tab作为分隔符的文件

UID	NAME	GENDER	BIRTHDATE
1	aaa	M	2012-11-12
2	bbb	M	2014-01-15
3	ddd	F	2014-07-05

首行字段

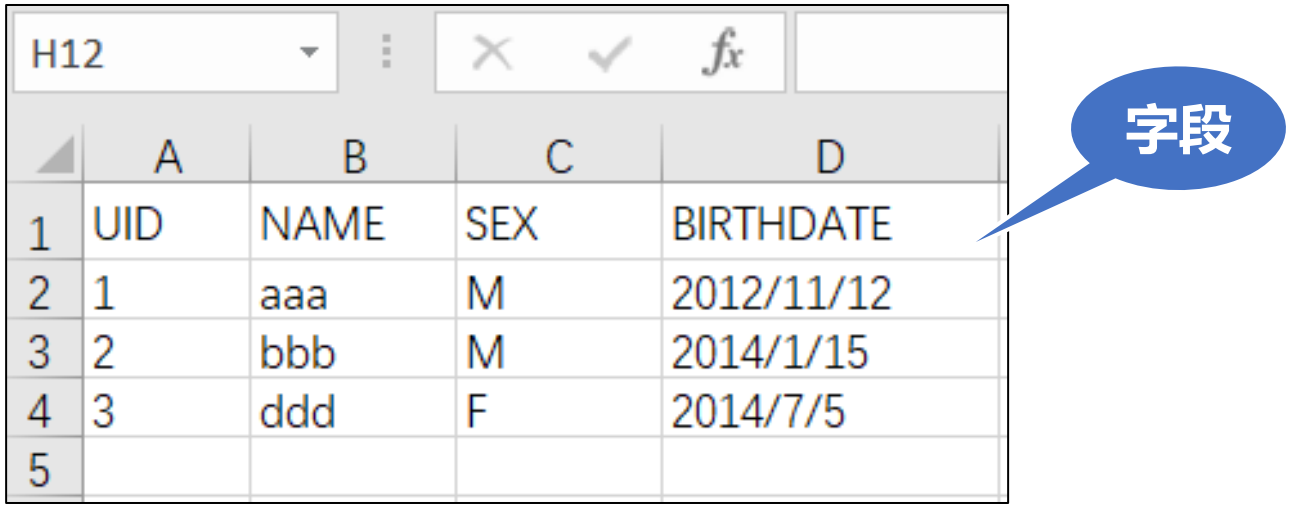
csv格式：常用逗号作为分隔符的文件

```
UID,NAME,GENDER,BIRTHDATE
1,aaa,M,2012-11-12
2,bbb,M,2014-01-15
3,ddd,F,2014-07-05
```

首行字段

10.3.1 数据文件

xls、xlsx： Microsoft Excel电子表格的文件格式



不同的文件格式的数据文件，都是序表数据的重要来源。

10.3.2 文件数据读取

文件数据读写时，导入可以使用f.import()，导出可以使用f.export()，这2个函数的选项@t表示导入/导出时第一行为字段名。

文件emp0.txt中首行带标题的数据如下：

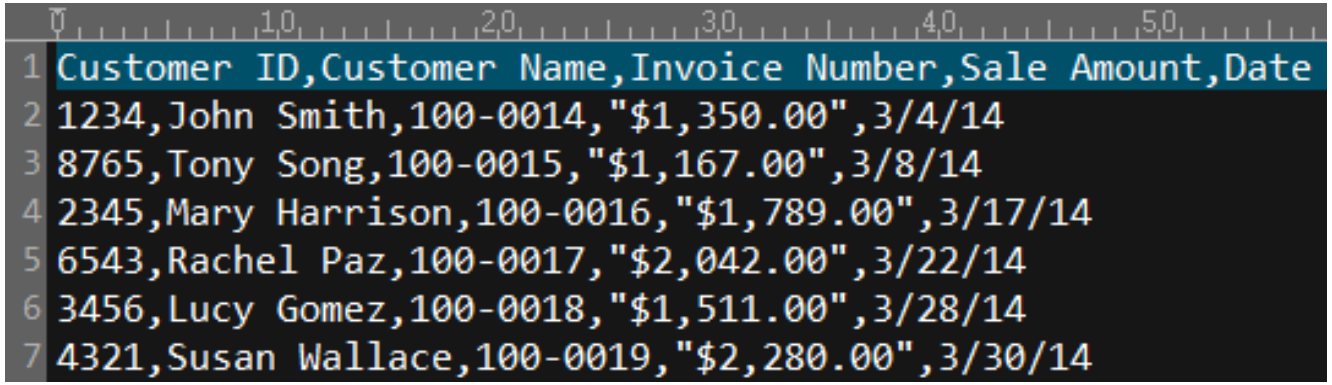
```
EID NAME  SURNAME GENDER  STATE BIRTHDAY  HIREDATE  DEPT  SALARY
201 Taylor  Brown F Georgia 1983-06-25  2004-03-01  2 5000
404 Madeline  Thomas  F Colorado  1977-01-08  2010-08-01  7 6500
203 Abigail Smith F Arizona 1972-02-20  2000-04-01  3 7000
205 Samantha  Moore F Washington  1977-10-10  2004-08-01  3 8000
402 Claire  Brown F California  1975-10-11  2010-07-01  7 7000
401 Olivia  Robinson  F Illinois  1971-04-03  2004-06-01  6 6500
210 Hannah  Johnson F Virginia  1984-07-09  2006-04-01  3 10000
398 Taylor  Smith F Pennsylvania  1986-12-13  2002-12-01  6 12000
397 Sarah Brown F Texas 1969-12-23  2005-12-01  3 6500
```

	A	B
1	=file("emp0.txt").import@t()	/使用@t选项，将第一行作为字段名

Index	EID	NAME	SURNAME	GENDER	STATE	…	SALARY
1	201	Taylor	Brown	F	Georgia	…	5000
2	404	Madeline	Thomas	F	Colorado	…	6500
3	203	Abigail	Smith	F	Arizona	…	7000
…	…	…	…				

10.3.2 文件数据读取

csv文件数据的访问。csv文件格式是常用“,”逗号作为数据之间的分隔符，文件sales.csv内容如下：



```
Customer ID,Customer Name,Invoice Number,Sale Amount,Date
1234,John Smith,100-0014,"$1,350.00",3/4/14
8765,Tony Song,100-0015,"$1,167.00",3/8/14
2345,Mary Harrison,100-0016,"$1,789.00",3/17/14
6543,Rachel Paz,100-0017,"$2,042.00",3/22/14
3456,Lucy Gomez,100-0018,"$1,511.00",3/28/14
4321,Susan Wallace,100-0019,"$2,280.00",3/30/14
```

	A	B
1	=file("sales.csv").import@tcq()	/读取csv文件数据

Index	Customer ID	Customer Name	Invoice Number	Sale Amount	Date
1	1234	John Smith	100-0014	$1,350.00	3/4/14
2	8765	Tony Song	100-0015	$1,167.00	3/8/14
3	2345	Mary Harrison	100-0016	$1,789.00	3/17/14
…	…	…	…	…	…

f.import@tcq()中的 @c 表示用逗号分隔符，@q 表示字段值外有引号则剥离。

10.3.2 文件数据读取

Excel文件导入用f.xlsimport()，导出用f.xlsexport()；
与f.import(), f.export()用法类似，当导入或导出与字段有关时，可用选项@t；Excel文件的访问如下：

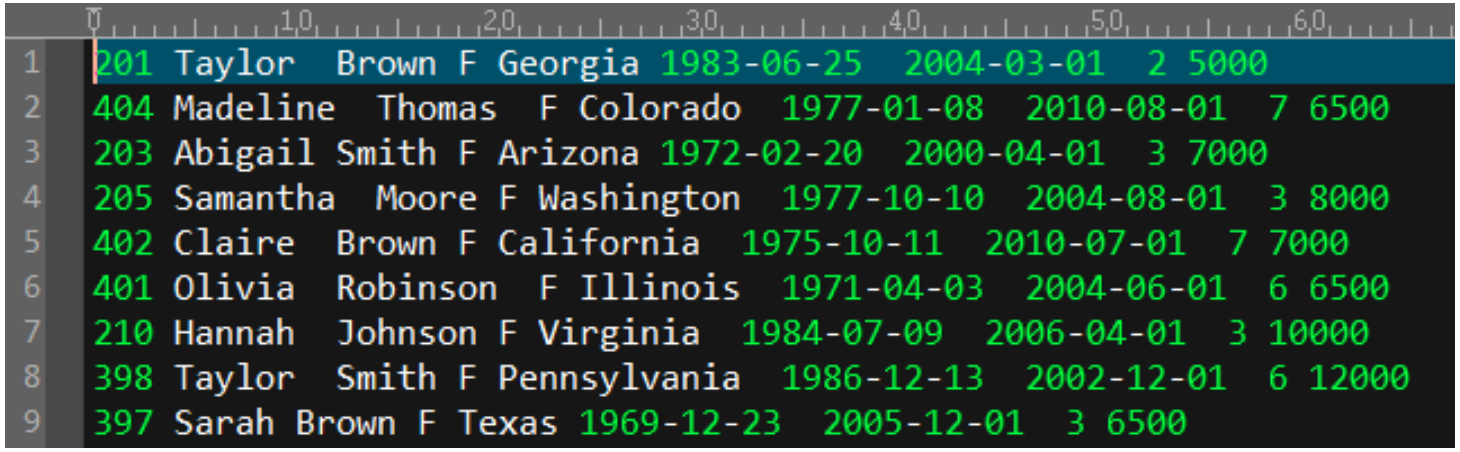
	A	B
1	=file("book_order.xlsx").xlsimport@t()	/读取xlsx文件数据

	A	B	C	D	E
1	PID	NAME	CODE	SIZE	DATE
2	9876	Daniel Farber	100-0008	1115	2013-02-02
3	8765	Laney Stone	100-0009	1367	2013-02-08
4	7654	Roger Lipney	100-0010	2135	2013-02-15
5	6543	Thomas Haines	100-0011		
6	5432	Anushka Vaz	100-0012		

Index	PID	NAME	CODE	SIZE	DATE
1	9876	Daniel Farber	100-0008	1115	2013-02-02
2	8765	Laney Stone	100-0009	1367	2013-02-08
3	7654	Roger Lipney	100-0010	2135	2013-02-15
4	6543	Thomas Haines	100-0011	1346	2013-02-17
5	5432	Anushka Vaz	100-0012	1560	2013-02-21

10.3.2 文件数据读取

无标题数据读取，生成序表的表头用_N表示字段名，其中N为正整数，无标题数据如下：



```
1  201 Taylor  Brown F Georgia 1983-06-25  2004-03-01  2 5000
2  404 Madeline  Thomas  F Colorado  1977-01-08  2010-08-01  7 6500
3  203 Abigail Smith F Arizona 1972-02-20  2000-04-01  3 7000
4  205 Samantha  Moore F Washington  1977-10-10  2004-08-01  3 8000
5  402 Claire  Brown F California  1975-10-11  2010-07-01  7 7000
6  401 Olivia  Robinson  F Illinois  1971-04-03  2004-06-01  6 6500
7  210 Hannah  Johnson F Virginia  1984-07-09  2006-04-01  3 10000
8  398 Taylor  Smith F Pennsylvania  1986-12-13  2002-12-01  6 12000
9  397 Sarah Brown F Texas 1969-12-23  2005-12-01  3 6500
```

	A	B
1	=file("emp0.txt").import()	/读入不带标题的文本数据

Index	_1	_2	_3	_4	_5	…	_9
1	201	Taylor	Brown	F	Georgia	…	5000
2	404	Madeline	Thomas	F	Colorado	…	6500
3	203	Abigail	Smith	F	Arizona	…	7000
4	205	Samantha	Moore	F	Washington	…	8000
…	…	…	…	…	…	…	…

10.3.2 文件数据读取

分隔符：对于文本数据使用特定分隔符，可在加载数据时指定。下面使用 “;” 作为文本数据之间的分隔符，emp0b.txt数据如下：

```
EID;NAME;SURNAME;GENDER;STATE;BIRTHDAY;HIREDATE;DEPT;SALARY
201;Taylor;Brown;F;Georgia;1983-06-25;2004-03-01;2;5000
404;Madeline;Thomas;F;Colorado;1977-01-08;2010-08-01;7;6500
203;Abigail;Smith;F;Arizona;1972-02-20;2000-04-01;3;7000
205;Samantha;Moore;F;Washington;1977-10-10;2004-08-01;3;8000
402;Claire;Brown;F;California;1975-10-11;2010-07-01;7;7000
401;Olivia;Robinson;F;Illinois;1971-04-03;2004-06-01;6;6500
210;Hannah;Johnson;F;Virginia;1984-07-09;2006-04-01;3;10000
398;Taylor;Smith;F;Pennsylvania;1986-12-13;2002-12-01;6;12000
397;Sarah;Brown;F;Texas;1969-12-23;2005-12-01;3;6500
```

	A	B
1	=file("emp0b.txt") .import@t(;,";")	/使用分号";"分隔符读入数据

返回内容与前面类似，不再列出。

f.import@t(fn$_1$,fn$_2$,...;k:n,s)中fn$_1$, fn$_2$,...表示要导入的字段，省略表示全字段导入；**k:n**表示将数据划分成**n**段只读第**k**段，省略时表示读取全文件；**s**表示分隔符，此处用分号。

10.3.2 文件数据读取

部分数据：有时并不需要加载所有数据，可以通过字段读入其中部分数据。

	A	B
1	=file("emp.txt")	
2	=A1.import@t(EID,NAME,BIRTHDAY,SALARY)	=A1.import@t(#1,#2,#6,#9)

Index	EID	NAME	BIRTHDAY	SALARY
1	1	Rebecca	1974-11-20	7000
2	2	Ashley	1980-07-19	11000
3	3	Rachel	1970-12-17	9000
4	4	Emily	1985-03-07	7000
…	…	…	…	…

读取数据时可以直接使用字段名，或用#i的方式指定字段序号，此处的#i分别对应字段为EID, NAME , BIRTHDAY, SALARY, 因此A2，B2中返回的数据相同。

10.3.2 文件数据读取

字段中带特殊字符：字段名称中间若有空格、￥，逗号等特殊字符，读取**指定字段**的数据则需要在字段名称两端加上单引号' '。 文件sales.csv内容如下：

```
Customer ID,Customer Name,Invoice Number,Sale Amount,Date
1234,John Smith,100-0014,"$1,350.00",3/4/14
8765,Tony Song,100-0015,"$1,167.00",3/8/14
2345,Mary Harrison,100-0016,"$1,789.00",3/17/14
6543,Rachel Paz,100-0017,"$2,042.00",3/22/14
3456,Lucy Gomez,100-0018,"$1,511.00",3/28/14
4321,Susan Wallace,100-0019,"$2,280.00",3/30/14
```

	A	B
1	=file("sales.csv").import@tcq('Customer ID','Customer Name', 'Invoice Number', 'Sale Amount')	/读取所给的字段数据

Index	Customer ID	Customer Name	Invoice Number	Sale Amount
1	1234	John Smith	100-0014	$1,350.00
2	8765	Tony Song	100-0015	$1,167.00
3	2345	Mary Harrison	100-0016	$1,789.00
…	…	…	…	…

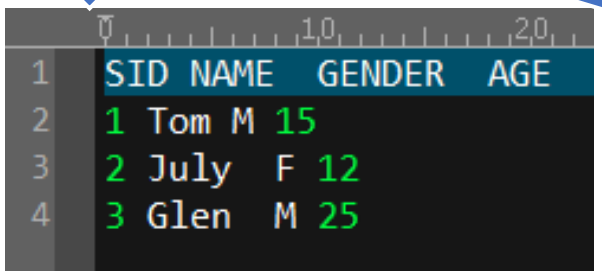
10.3.3 文件数据的写出

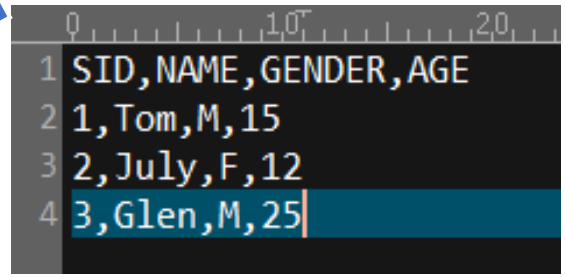
序表读写到txt和csv文件，保存到本地文件中。

	A	B	C	D
1	1	Tom	M	15
2	2	July	F	12
3	3	Glen	M	25
4	=create(SID,NAME, GENDER, AGE)	>A4.record([A1:D3])	/创建序表，填入数据	
5	>file("txt/user_100.txt").export@t(A4)		/将表数据保存为user_100.txt	
6	>file("txt/user_200.csv").export@tc(A4)		/将表数据保存为user_200.csv	

A4返回的数据：

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	12
3	3	Glen	M	25





f.export@t()将数据输出存储到文件中，其中@t表示将字段名作为首条记录写入文件，缺省采用tab为分隔符，用@c表示采用逗号为分隔符。

10.3.3 文件数据的写出

修改的数据输出到文件中。

	A	B	C	D
1	1	Tom	M	15
2	2	July	F	12
3	3	Glen	M	25
4	=create(SID,NAME, GENDER, AGE)	>A4.record([A1:D3])	/序表	
5	>A4(1).GENDER="Male"		/修改第1条记录GENGER字段的值	
6	>A4(2).GENDER="Female"		/修改第2条记录GENGER字段的值	
7	>A4(3).GENDER="Male"		/修改第3条记录GENGER字段的值	
8	>file("txt/user_101.txt").export@t(A4)		/将表数据保存为user_101.txt	

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	12
3	3	Glen	M	

修改

Index	SID	NAME	GENDER	AGE
1	1	Tom	Male	15
2	2	July	Female	12
3	3	Glen	Male	25

A8导出

```
SID NAME  GENDER  AGE
1 Tom Male  15
2 July  Female  12
3 Glen  Male  25
```

10.3.3 文件数据的写出

增加的数据输出到文件中。

	A	B	C	D
1	1	Tom	M	15
2	2	July	F	12
3	3	Glen	M	25
4	=create(SID,NAME, GENDER, AGE)	>A4.record([A1:D3])	/序表	
5	>A4.insert(0, 4, "Tina", "F":GENDER, 10)		/增加1条数据	
6	>A4.insert(0, 5, "Alisa", "F", 11)		/增加1条数据	
7	>file("txt/user_102.txt").export@t(A4)		/将表数据保存为user_102.txt	

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	
3	3	Glen	M	

A5、A6插入

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	12
3	3	Glen	M	25
4	4	Tina	F	10
5	5	Alisa	F	11

A7导出

```
SID NAME  GENDER  AGE
1 Tom M 15
2 July  F 12
3 Glen  M 25
4 Tina  F 10
5 Alisa F 11
```

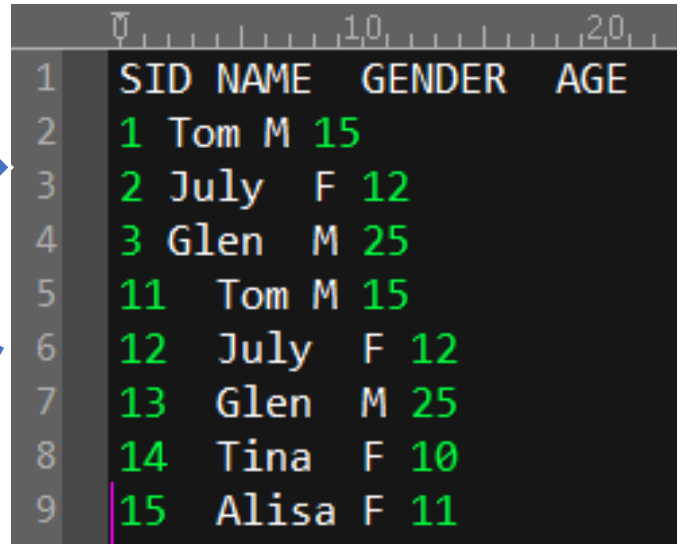
10.3.3 文件数据的写出

文件合并输出，f.export@a(T)将结构相同的数据追加到已有文件之后

	A	B
1	=file("txt/user_102.txt").import@t()	/数据导入到序表
2	>file("txt/user_101.txt").export@ta(A1)	/将表数据追加方式保存到已有文件user_101.txt

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	12
3	3	Glen	M	25

Index	SID	NAME	GENDER	AGE
1	11	Tom	M	15
2	12	July	F	12
3	13	Glen	M	25
4	14	Tina	F	10
5	15	Alisa	F	11



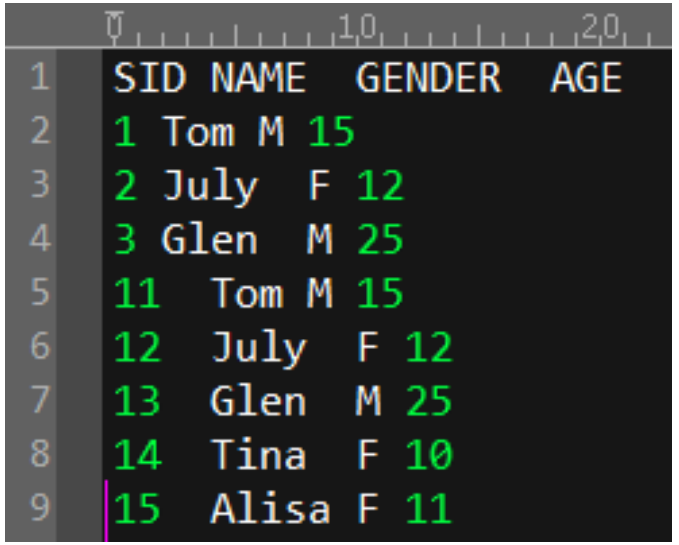
10.3.3 文件数据的写出

文件合并输出，先合并数据再写出

	A	B	C
1	=file("txt/user_100.txt").import@t()		
2	=file("txt/user_102.txt").import@t()		
3	for A2	>A1.insert@r(0:[A3])	/将表A2的数据合并到表A1中
4	>file("txt/user_103.txt").export@t(A1)		/将表数据保存为user_103.txt

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	12
3	3	Glen	M	25

Index	SID	NAME	GENDER	AGE
1	11	Tom	M	15
2	12	July	F	12
3	13	Glen	M	25
4	14	Tina	F	10
5	15	Alisa	F	11



10.3.3 文件数据的写出

文件拆分输出，根据条件将文件数据先分开，再输出到不同的文件中。

	A	B	C
1	=file("txt/user_103.txt").import@t()		
2	=A1.create()	=A1.create()	
3	for A1	if A3.SID<10	
4			=A2.insert@r(0:[A3])
5		else	
6			=B2.insert@r(0:[A3])
7	>file("txt/user_110.txt").export@t(A2)		
8	>file("txt/user_111.txt").export@t(B2)		

Index	SID	NAME	GENDER	AGE
1	1	Tom	M	15
2	2	July	F	12
3	3	Glen	M	25
4	11	Tom	M	15
5	12	July	F	12
6	13	Glen	M	25
7	14	Tina	F	10
8	15	Alisa	F	11

A7导出

```
SID NAME  GENDER  AGE
1 Tom M 15
2 July  F 12
3 Glen  M 25
```

A8导出

```
SID NAME  GENDER  AGE
11  Tom M 15
12  July  F 12
13  Glen  M 25
14  Tina  F 10
15  Alisa F 11
```

10.3.4 数据库DBMS

关系数据库是当前最重要的结构化数据存储场所，我们操作的许多结构化数据都来自且要写回关系数据库。

结构化查询语言SQL是关系数据库上管理数据的一种语言。

可以利用SQL从关系数据库中取出数据：

```
mysql> select * from emp where eid<10;
+------+----------+---------+--------+------------+------------+------------+-----------+--------+--------------+
| EID  | NAME     | SURNAME | GENDER | STATE      | BIRTHDAY   | HIREDATE   | DEPT      | SALARY | IDENTITYCARD |
+------+----------+---------+--------+------------+------------+------------+-----------+--------+--------------+
|    1 | Rebecca  | Moore   | F      | California | 1974-11-20 | 2005-03-11 | R&D       |   7000 | 1230001      |
|    2 | Ashley   | Wilson  | F      | New York   | 1980-07-19 | 2008-03-16 | Finance   |  11000 | 1230002      |
|    3 | Rachel   | Johnson | F      | New Mexico | 1970-12-17 | 2010-12-01 | Sales     |   9000 | 1230003      |
|    4 | Emily    | Smith   | F      | Texas      | 1985-03-07 | 2006-08-15 | HR        |   7000 | 1230004      |
|    5 | Ashley   | Smith   | F      | Texas      | 1975-05-13 | 2004-07-30 | R&D       |  16000 | 1230005      |
|    6 | Matthew  | Johnson | M      | California | 1984-07-07 | 2005-07-07 | Sales     |  11000 | 1230006      |
```

关系数据库的知识相对专业，也有大量的参考书籍，本文就不再赘述

10.4 序表的运算

序表的运算

- 序表也是个序列，可以使用第8章中讲过的序列函数，如使用**for T**循环序表中的**每条记录**。
- 类似序列，在循环序表时，也可以针对所有成员统计出**聚合**结果，如**计数、求和、求平均值，求极值**等。但是序列在聚合是对序列中的成员执行统计，序表在聚合时经常针对**某个字段的值**对**所有记录**执行统计。
- 在循环序表时，还可能需要对记录执行**过滤**，选出符合条件的记录添加到同结构的新序表中。

10.4.1 聚合

计数，求当前用户数。通过循环累计表中的用户数量。

	A	B	C	D
1	=file("txt/user_1.txt").import@t()	0		/B1计数所有用户数
2	>iM=0			/iM计数男性数
3	for A1	>B1=B1+1		/累计用户个数
4		if A3.GENDER=="M"		/判断是否为男性
5			>iM=iM+1	/累计男性个数
6	>output("count=" / B1/"; iM=" / iM)			/输出用户数、男性个数

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

输出：count = 4; iM=3

A3执行for循环，循环每一位用户。在循环时，B3累计用户个数，C5累计男性个数，直到循环结束。

序表实际上也是**记录构成的序列**，因此用**for T**循环序表，和第8章中介绍的用for A循环序列本质上是相同的。只不过序表中，每个成员都是记录，如上面代码中，执行循环时，循环主格A3中的值就依次为各条记录。而在序表的循环计算时，通常需要对记录的某个字段值进行统计。

10.4.1 聚合

求和，求所有男性用户年龄和。循环中计算出每位男性用户年龄，再累计年龄。

	A	B	C	D
1	=file("txt/user2.txt").import@t()	0		/B1累计年龄数
2	for A1	if A2.GENDER=="M"		/判断是否为男性
3			=age(A2.BIRTHDAY)	/计算用户年龄大小
4			>B1=B1+C3	/累计年龄
5	>output("sum=" / B1)			/输出总年龄

Index	SID	NAME	GENDER	BIRTHDAY
1	1	Jack	M	2007-05-03
2	2	Tom	M	2004-07-12
3	3	July	F	2010-09-04
4	4	Gail	M	2010-03-08

输出：sum=39

在循环时，先根据用户出生日期计算出年龄，再累计年龄。

10.4.1 聚合

求平均值。下面求用户平均年龄，将累计所有用户年龄之和及用户数，然后用累计年龄除以用户数计算出平均年龄。

	A	B	C
1	0	0	/B1累计年龄数
2	=file("txt/user2.txt").import@t()		/读入文本文件数据
3	for A2	=age(A3.BIRTHDAY)	/计算用户年龄大小
4		>B1=B1+B3	/累计年龄
5		>A1=A1+1	/累计用户数
6	=round(B1/A1, 2)		/计算平均年龄
7	>output("avg="/A6)		/输出平均年龄

输出：avg=12.0

for T 循环每一位用户记录，先根据用户出生日期计算出年龄，再累计用户年龄及用户数，循环结束后，用累计年龄除以用户数就计算出平均年龄，round()四舍五入并保留两位小数。

10.4.1 聚合

求极值。计算用户最大、最小年龄。

	A	B	C	D
1	>iMax=0	>iMin=99		/初始化最大、最小年龄参数
2	=file("txt/user_2.txt").import@t()			/读入文本文件数据
3	for A2	=age(A3.BIRTHDAY)		/计算用户年龄大小
4		if B3>iMax	>iMax=B3	/比较得到较大年龄
5		if B3<iMin	>iMin=B3	/比较得到较小年龄
6	>output("Min="/iMin /"; Max="/iMax)			/输出最大、最小年龄

输出： Min=9； Max=16

在for T循环时，先根据用户出生日期计算出年龄，再比较出较大年龄、较小年龄分别存储到变量iMax, iMin中，循环结束后就得到用户的最大、最小年龄。

10.4.1 聚合

Top N，计算5个最大年龄值

序列B1用来存储5个最大年龄值，初始化值全为0。
循环遍历用户记录时，外循环根据出生日期计算出年龄，在内循环用它与序列B1中的成员依次比较，若年龄比B1中的大，则在此位置插入年龄，同时删除最后一个值，中断内循环；如此反复，直到循环结束，由此计算出5个年龄最大值。

	A	B	C	D	E
1	=file("txt/emp.txt").import@t()	[0,0,0,0,0]			/B1插入5条空记录
2	for A1	=age(A2.BIRTHDAY)			/初始化出生日期
3		for 5	if B2>B1(B3)	>B1.insert(B3, B2)	/插入比较大的值
4				>B1.delete(6)	/删除最后一个值
5				break	

10.4.2 过滤

将符合条件的记录存入序表，对于同结构记录，可用T.insert@r(0:[r])：

	A	B	C
1	=file("txt/employee.txt").import@t()	=A1.create()	/生成同结构的序表
2	for A1	if A2.BIRTHDAY>date("1987-08-01")	/查找出生日期大于1987-08-01的记录
3			>B1.insert@r(0:[A2])

C3：将符合条件的记录插入到B1中

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	…
1	1	Rebecca	Moore	F	California	1974-11-20	…
2	2	Ashley	Wilson	F	New York	1980-07-19	…
3	3	Rachel	Johnson	F			
4	4	Emily	Smith	F			
5	5	Ashley	Smith	F			
…	…	…	…				

B1序表

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	…
1	89	Emily	Johnson	F	Michigan	1987-12-09	…
2	241	Samantha	Davis	F	Georgia	1987-12-04	…
3	393	Hannah	Johnson	F	Florida	1987-09-06	…
4	437	Zachary	Williams	M	Tennessee	1987-12-01	…
5	472	Hannah	Brown	F	California	1987-12-27	…

10.4.2 过滤

将符合条件的记录存入序表，对于与序表有相同字段的记录，则插入记录时将匹配的字段名对应的值写入序表，可用T.insert@f(0:[r])：

	A	B	C
1	=file("txt/employee.txt").import@t()	=create(EID, NAME, STATE, BIRTHDAY)	/B1过滤后用于记录存储的序表
2	for A1	if A2.BIRTHDAY>date("1987-01-01") && A2.STATE=="California"	/查找出生日期大于1987-01-01并且州名为California的记录
3			>B1.insert@f(0:[A2])

C3：将同时符合所给两个条件的记录插入到表B1中

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	…
1	1	Rebecca	Moore	F	California	1974-11-20	…
2	2	Ashley	Wilson	F	New York	1980-07-19	
3	3	Rachel	Johnson	F	New Mexico		
4	4	Emily	Smith	F	Texas		
5	5	Ashley	Smith	F	Texas		
…	…	…	…				

B1序表

Index	EID	NAME	STATE	BIRTHDAY
1	119	Madison	North Caro…	1987-05-10
2	123	Ashley	Indiana	1987-02-09
3	151	Samantha	California	1987-02-06
…	…	…	…	…

10.4.2 过滤

使用表达式过滤。下面查找“Ca”开头用户名的记录：

	A	B	C
1	=file("txt/employee.txt").import@t()	=A1.create()	/创建同结构的新序表
2	for A1	if left(A2.NAME,2)=="Ca"	/查找用户名“Ca”开头的的记录
3			>B1.insert@r(0: [A2])

B2：函数left(A2.NAME,2)截取用户名前两个字符的子串，若子串等于“Ca”，则将记录追加到序列B1中

Index	EID	NAME	SURNAME	GENDER	STATE	…	SALARY
1	1	Rebecca	Moore	F	California	…	7000
2	2	Ashley	Wilson	F	New York	…	11000
3	3	Rachel	Johnson	F	New Mexico	…	9000
4	4	Emily	Smith	F	Texas	…	7000
…	…	…	…				

B1数据：

Index	EID	NAME	SURNAME	GENDER	STATE	…	SALARY
1	67	Cameron	Jones	M	New York		5000
2	220	Caleb	Smith	M	New York		7000
3	289	Cameron	Johnson	M	California		7000

10.5 排列

10.5.1 排列概念

上节最后一个例子，查找“ Ca ”开头用户名的记录，我们换一种写法来实现：

	A	B	C
1	=file("txt/employee.txt").import@t()	[]	/定义空序列B1
2	for A1	if left(A2.NAME,2)=="Ca"	/查找用户名“Ca”开头的的记录
3			>B1=B1|A2

B2：截取用户名子串，若子串等于“Ca”，则将记录追加到序列B1中

Index	EID	NAME	SURNAME	GENDER	STATE	…	SALARY
1	67	Cameron	Jones	M	New York		5000
2	220	Caleb	Smith	M	New York		7000
3	289	Cameron	Johnson	M	California		7000

在B2中使用**空序列**而没有通过A1.create()创建空序表，在C3中在序列中添加成员，而没有使用序表的插入函数，结果得到的实际上是**记录构成的序列**，而不是上一节中的新序表，但是查看B2的值时，显示效果和序表是类似的。

10.5.1 排列概念

从这个例子可以看到，多条记录构成的序列，看起来和序表很像。

序表本身也是记录构成的序列，而这些记录可以再用来构成新序列，这个新序列我们称为排列。即，**排列**就是**记录构成的序列**，由同一序表的记录构成的序列则称为纯排列。

序表有数据结构，**普通的排列则没有数据结构**，但是**所有记录数据结构相同的排列（如纯排列）**能够借用**记录的数据结构**来显示。序表通过数据结构来处理数据，而那些不通过数据结构处理数据的结果不再是序表，譬如上一页的合集结果是排列而非序表。

实际上，序表也是全部由记录构成的序列，因此在运算中，也可以把序表视为一种**特殊的排列**。

10.5.1 排列概念

排列是记录构成的序列。

下面演示排列的组成，其数据成员是来自序表的记录：

序表A数据

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

1:[2, Tom, M, 15]

2:[1, Jack, M, 12]

3:[4, Gail, M, 11]

排列数据

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	15
2	1	Jack	M	12
3	4	Gail	M	11

从序表中抽取出记录A([2,1,4])来组成排列。

10.5.1 排列概念

直接从序表中取出记录构成排列：

	A	B
1	=file("txt/user_1.txt").import@t()	/导入数据, 返回序表
2	=A1([2, 1, 4])	/用序表中的第2、1、4条记录组成排列

序表A1数据：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

排列A2数据：

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	15
2	1	Jack	M	12
3	4	Gail	M	11

10.5.1 排列概念

数据修改的影响：将序表数据复制到新序表与排列，然后修改源序表的记录，观察对新序表、排列的影响，操作如下：

	A	B	C
1	=file("txt/user_4.txt").import@t()		/导入数据
2	=A1.create()	[]	/生成新序表A2，定义空序列B2
3	for A1	>A2.insert@r(0:[A3])	/插入数据到序表A2
4		>B2=B2|A3	/数据合并到排列B2
5	=A1(2).NAME="July33"		/修改源序表第2行记录NAME值

序表A2数据：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	3	July	F	11

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	3	July	F	11

排列B2数据：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	3	July	F	11

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	3	July33	F	11

A2中的新序表只是复制了A1的结构，使用A.insert()时都是**新建的记录**，与A1中记录无关，所以A1修改时不会影响A2。排列B2中的第2条记录NAME值变成July33了，因为排列并不存储实体记录，而是**记录对象**，因此A1中序表记录修改会导致排列记录也变化。

10.5.1 排列概念

当记录被修改时，内存中所有引用同一记录的变量也都会看到变化，这就是排列的**引用属性**。

	A	B
1	=file("txt/user_1.txt").import@t()	/序表
2	=[A1(4), A1(1), A1(3), A1(2), A1(1)]	/构造带重复记录的排列
3	>A1.modify(1, 10, "JACKER")	/对序表首条记录修改。
4	=A2(2)==A2(5)	/返回true

A1	SID	NAME	GENDER	AGE
	1	Jack	M	12
	2	Tom	M	15
	3	July	F	11
	4	Gail	M	13

A3修改

A1	SID	NAME	GENDER	AGE
	10	JACKER	M	12
	2	Tom	M	15
	3	July	F	11
	4	Gail	M	13

A2	SID	NAME	GENDER	AGE
	4	Gail	M	13
	1	Jack	M	12
	3	July	F	11
	2	Tom	M	15
	1	Jack	M	12

A3修改

A2	SID	NAME	GENDER	AGE
	4	Gail	M	13
	10	JACKER	M	12
	3	July	F	11
	2	Tom	M	15
	10	JACKER	M	12

10.5.1 排列概念

排列的记录还可以来自多个序表。

序表A数据

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

序表B数据

Index	SID	NAME	GENDER	AGE
1	13	Glen	M	25
2	14	Tina	F	10
3	15	Alisa	F	11

排列数据

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	13	Glen	M	25
5	14	Tina	F	10
6	15	Alisa	F	11

10.5.1 排列概念

甚至来自不同结构的多个序表：

序表A数据

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

序表B数据

Index	SID	NAME	GENDER	BIRTHDAY
1	13	Glen	M	1985-02-06
2	14	Tina	F	2010-04-08
3	15	Alisa	F	2005-06-03

排列数据

Index	SID	NAME	GENDER	AGE	BIRTHDAY
1	1	Jack	M	12	(null)
2	2	Tom	M	15	(null)
3	3	July	F	11	(null)
4	13	Glen	M	(null)	1985-02-06
5	14	Tina	F	(null)	2010-04-08
6	15	Alisa	F	(null)	2005-06-03

这里只是为了**演示**排列中的数据，如果排列中记录的数据结构不同，实际上无法类似序表一样同时按数据结构显示。

10.5.1 排列概念

数据成员重复性，同一记录可以**重复**地出现在排列中：

序表数据

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

1:[1, Jack, M, 12]
2:[2, Tom, M, 15]
3:[3, July, F, 11]
4:[2, Tom, M, 15]
5:[2, Tom, M, 15]
6:[1, Jack, M, 12]
7:[1, Jack, M, 12]

排列数据

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	2	Tom	M	15
5	2	Tom	M	15
6	1	Jack	M	12
7	1	Jack	M	12

10.5.1 排列概念

排列数据可以来自不同的序表，不仅指直接来自序表的记录，也指间接来自序表的记录，即来自排列中的记录。

	A	B	C
1	=file("txt/user_1.txt").import@t()		/序表
2	=file("txt/user_3.txt").import@t()		/序表
3	=[A1(4), A1(1), A1(3), A2(1)]	=[A1(1), A1(3), A2(2), A2(2), A3(4)]	/构造不同的排列

序表A1

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	13

序表A2

Index	SID	NAME	GENDER	AGE
1	20	Liky	F	9
2	21	Lucy	F	7
3	22	Mark	M	13

排列A3

Index	SID	NAME	GENDER	AGE
1	4	Gail	M	13
2	1	Jack	M	12
3	3	July	F	11
4	20	Liky	F	9

排列B3

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	3	July	F	11
3	21	Lucy	F	7
4	21	Lucy	F	7
5	20	Liky	F	9

10.5.2 排列的运算

排列的生成，向定义的空序列合并记录，或通过记录构造。

	A	B
1	=file("txt/user_1.txt").import@t()	/序表
2	[]	/定义空序列
3	>A2=A2|A1(2)	/合并1条记录
4	>A2=A2|A1([1, 3])	/合并多条记录
5	=[A2(2), A2(1), A2(3), A2(3)]	/通过排列中的记录构建新排列
6	=A2([2, 1, 3, 3])	/通过序号数列获取排列中记录构建新排列
7	=A1.m([2, 3, 4])	/A.m()，通过序号获取记录构建排列

A2排列：

Index	Member

A3合并

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	15

A4合并

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	15
2	1	Jack	M	12
3	3	July	F	11

A5与A6排列：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	3	July	F	11

A7排列：

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	15
2	3	July	F	11
3	4	Gail	M	11

10.5.2 排列的运算

序表和排列都是由记录构成的序列，对于**排列的生成**，使用函数**A.to()**有时更便捷。

	A	B
1	=file("txt/user_1.txt").import@t()	/导入数据
2	=A1.to()	/序表A1中所有记录组成的排列
3	=A1.to(2)	/取前2条记录组成排列
4	=A1.to(2,4)	/从表第2行到第4行之间的记录组成的排列
5	=A1.to(,3)	/从表首行到第3行之间的记录组成的排列

A2：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

A3：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15

A4：

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	15
2	3	July	F	11
3	4	Gail	M	11

A5：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11

10.5.2 排列的运算

排列作为**序列**，可像序列一样进行访问、更新、删除等运算。

	A	B	C
1	=file("txt/user_1.txt").import@t()	0	/序表
2	=[A1(1), A1(4)]		/用记录构造排列
3	=A1.to()		/A.to()方式生成排列
4	=A3(1)		/访问首行记录。
5	=A3.m(1)		/同上
6	=A3.m(-1)		/访问最后一条记录
7	>A2(2)=A4		/记录更新
8	>A2.insert(0, A4)		/插入记录。
9	>A2.delete(1)		/删除记录
10	=A2|A3		/合集
11	=A2&A3		/并集
12	=A2^A3		/交集
13	for A3	>B1=B1+1	/遍历访问

10.5.2 排列的运算

排列本身是没有绑定数据结构的，但它的成员是记录，因此对排列还可以用r.F或r.#i方式来访问记录的字段值，与序表记录的访问类似。

	A	B	C
1	=file("txt/user_1.txt").import@t()		/B1定义空表
2	=A1.to()		/生成排列
3	=A2(1)		/访问排列首条记录
4	=A3.NAME	=A3.#2	/访问NAME值
5	=A2(1).NAME	=A2(1).#2	/访问NAME值
6	>A2(1).NAME="Jacker"	>A2(1).#2="Jacker"	/修改NAME值
7	>A2(1).GENDER="Male"	>A2(1).#3="Male"	/修改GENDER值

10.5.2 排列的运算

排列的**循环运算**和序表类似。示例：计数，求用户数。

	A	B	C	D
1	=file("txt/user_1.txt").import@t()	0		/B1计数所有用户数
2	=A1.to()	>iM=0		/iM计数男性数
3	for A2	>B1=B1+1		/累计用户个数
4		if A3.GENDER=="M"		/判断是否为男性
5			>iM=iM+1	/累计男性个数
6	>output("count=" / B1/"; iM=" / iM)			/输出用户数、男性个数

求和，求所有男性用户年龄和。

	A	B	C	D
1	=file("txt/user2.txt").import@t()	=A1.to()	0	/C1累计年龄总和
2	for B1	if A2.GENDER=="M"		/判断是否为男性
3			=age(A2.BIRTHDAY)	/计算用户年龄大小
4			>C1=C1+C3	/累计年龄
5	=output("sum=" / C1)			/输出总年龄

排列同样是记录构成的序列，排列的循环和序表是**相同**的。

10.5.2 排列的运算

求平均值，下面求用户平均年龄。

	A	B	C
1	0	0	/B1累计年龄数
2	=file("txt/user2.txt").import@t()	=A2.to()	/通过序表生成排列
3	for B2	=age(A3.BIRTHDAY)	/计算用户年龄大小
4		>B1=B1+B3	/累计年龄
5		>A1=A1+1	/累计用户数
6	=round(B1/A1, 2)		/计算平均年龄
7	=output("avg="/A6)		/输出平均年龄

求极值，求用户最大、最小年龄。

	A	B	C	D
1	>iMax=0	>iMin=99		/初始化最大、最小年龄参数
2	=file("txt/user_2.txt").import@t()	=A2.to()		/通过序表生成排列
3	for B2	=age(A3.BIRTHDAY)		/计算用户年龄大小
4		if B3>iMax	>iMax=B3	/比较得到较大年龄
5		if B3<iMin	>iMin=B3	/比较得到较小年龄
6	=output("Min="/iMin /"; Max="/iMax)			/输出最大、最小年龄

10.5.2 排列的运算

条件过滤，将符合条件的记录存入排列

	A	B	C
1	=file("txt/employee.txt").import@t()	[]	/B1为空列，用于存储过滤后的记录
2	for A1	if A2.BIRTHDAY>date("1987-01-01") && A2.STATE=="California"	/查找出生日期大于1987-01-01并且州名为California的记录
3			>B1=B1|A2

使用表达式过滤。下面查找以“Ca”开头用户名的记录：

	A	B	C
1	=file("txt/emp.txt").import@t()	[]	/定义空列
2	for A1	if left(A2.NAME,2)=="Ca"	/查找用户名以“Ca”开头的的记录
3			>B1=B1|A2

将过滤后的记录存入排列，下面用like()查找以“son”结尾的用户名记录：

	A	B	C
1	=file("txt/emp.txt").import@t()	[]	/定义空列
2	for A1	if like(A2.NAME,"*son")	/查找用户名以“son”结尾的的记录
3			>B1=B1|A2

10.5.2 排列的运算

TopN，求年龄最大的5个用户

初始化排列时，将前5个用户中出生日期最大的，即年龄最小的记录作为B1排列的成员。
循环遍历用户记录时，将较小出生日期的记录按由小到大顺序插入排列B1，同时删除最后一条记录；如此反复，直到循环结束，由此计算出5个年龄最大值。

	A	B	C	D	E
1	=file("txt/emp.txt").import@t()	=A1([4,4,4,4,4])			/用户记录构成排列
2	for A1	for 5	if A2.BIRTHDAY<B1(B2).BIRTHDAY	>B1.insert(B2,A2)	/将较小出生日期记录插入
3				>B1.delete(6)	/删除最后一条记录
4				break	

通过聚合运算与条件过滤，我们看到排列与序表在使用方式上类似。

10.5.2 排列的运算

排列可以执行集合运算，下面两个排列数据均来自同一序表，其中[2,3]条记录是共同的，然后进行合集、交集、并集、差集运算。

	A	B	C
1	=file("txt/user_1.txt").import@t()		/导入数据
2	=A1([1, 2, 3])	=A1([2, 3, 4])	/生成排列
3	=A2|B2		/排列间合集，返回集合中的所有记录
4	=A2^B2		/排列间交集，返回共同记录
5	=A2&B2		/排列间并集，返回去重后的记录
6	=A2\B2		/排列间差集，A2中有但B2中无的记录

A2|B2数据：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	2	Tom	M	15
5	3	July	F	11
6	4	Gail	M	11

A2\B2数据：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12

A2^B2数据：

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	15
2	3	July	F	11

A2&B2数据：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

10.5.2 排列的运算

源自不同序表的排列进行交集运算。下面两个排列来自有相同数据的两个序表。

	A	B	C
1	=file("txt/user_1.txt").import@t()	=A1.create()	/建立新序表B1
2	>B1.insert@r(0:A1)		/数据复制到新序表B1
3	=A1([1, 2, 3])	=B1([1, 2, 3])	/生成两个排列
4	=A3^B3		/排列间交集，返回空

A3^B3数据：

Index	Member

A4：两个排列的交集返回为空。

不同序表的成员总是不同的，即使它们的字段及值都相同，因此源自不同序表的排列进行交集运算总是返回空，只有源自同一序表的排列进行集合运算才有意义。

10.5.2 排列的运算

应用排列交集“^”找出符合过滤条件的记录。下面查询2015年期间销售额在20000以上的数据。

	A	B	C	D
1	=file("txt/sales.txt").import@t()	[]	[]	/导入数据
2	for A1	if A2.AMOUNT>20000	>B1=B1|A2	/将销售额在20000以上的数据存入排列B1
3		if A2.ORDERDATE>=date("2015-01-01") && A2.ORDERDATE<=date("2015-12-31")	>C1=C1|A2	/将2015年期间的数据存入排列C1
4	=B1^C1			/计算相同记录

A4数据：

Index	ORDERID	CLIENT	SELLERID	AMOUNT	ORDERDATE
1	790	HANAR	13	28200.0	2015-01-04 15:28:05
2	800	EGU	20	29400.0	2015-01-10 15:28:05
3	802	QUICK	8	28600.0	2015-01-13 15:28:05
4	806	MIP	12	20900.0	2015-01-14 15:28:05
…	…	…	…		

通过循环，查找出20000以上的销售单与2015年的销售单分别存入排列B1、C1，循环结束后再求B1、C1交集，即为所要的结果数据。

10.5.2 排列的运算

如果要比较不同序表中记录是否字段值相同怎么办?

在表A、B中找出有重复字段值的记录，我们可用cmp(r_1,r_2)进行记录值比较找出相同记录。
从A中取出一条记录与B中记录依次比较，若相等则将此记录存入排列；反复此过程，即可找出字段值相同的记录。

	A	B	C	D	E
1	=file("txt/sales.txt").import@t()	=A1.create()	=A1.create()		/导入数据
2	for A1	if A2.AMOUNT>20000	>B1.insert@r(0:[A2])		/将销售额在20000以上的数据存入序表B1
3		if A2.ORDERDATE>=date("2015-01-01") && A2.ORDERDATE<=date("2015-12-31")	>C1.insert@r(0:[A2])		/将2015年期间的数据存入序表C1
4	for B1	for C1	if cmp(A4,B4)==0	>D1=D1|A4	break

计算结果与前面排列交集运算完全相同。

cmp(r_1,r_2)当两条记录r_1,r_2比较大小时，两条记录的所有字段值完全相等返回0。

10.5.3 排列与序表的不同

序表有数据结构，而**排列本身是没有数据结构的**，有些序表的运算不被支持：

	A	B
1	=file("txt/user_1.txt").import@t()	/导入文本数据
2	=A1.to()	/生成排列
3	>B1.insert(2:A1(3))	/插入记录
4	/=B1.insert(1, 11:SID)	/不支持
5	>A1.modify(2, "Tom22":NAME)	/修改序表第2行NAME为Tom22
6	>A2(1).NAME="Jacker"	/修改排列第1行NAME为Jacker
7	>A1.delete(1)	/删除序表中的第1条记录, 不影响排列数据
8	>A2.delete(3)	/删除排列中的第3行记录

1、排列增加记录时，**不支持**A.insert(k,x:F), A.insert(k:A,x:F), A.insert@r(k:A), A.insert@f(k:A) 。

2、排列修改数据时，**不支持**A.modify(k,x:F), A.modify(k:A,x :F), A.modify@r, A.modify@f。

3、删除序表中的记录不会影响排列中的数据。

4、排列也**不能**使用A.record() 填充数据

10.5.3 排列与序表的不同

由于排列也是一种序列，可向排列中增加非记录数据，若再把它当作排列操作就有可能出错。

	A	B	C
1	=file("txt/user_1.txt").import@t()		/序表
2	=A1.to()		/生成排列
3	>A2.insert(1,11)	=A2(1)	/插入非记录数据
4	/=B3.NAME	/=B3.#2	/访问NAME值
5	=A2(2).NAME	=A2(2).#2	/访问NAME值

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	15
2	1	Jack	M	12
3	3	July	F	11

A3插入数据

Index	Member
1	11
2	[2, Tom, M, …]
3	[1, Jack, M, …]
4	[3, July, F…]

B3没有字段信息，不能当作排列访问了。针对排列，尽量避免类似这样的操作。

10.5.3 排列与序表的不同

排列可由来自不同结构的序表记录构成，但只能进行有限的计算。

下面统计来自两个部门A、B的人员工资，其中部门B提供的表结构多了“补贴”字段。
由于工资字段是两个序表都有的，可将序表数据先合并成排列，再遍历排列记录，累计工资。

	A	B	C
1	=file("txt/emp_c1.txt").import@t()		/加载emp_c1数据
2	=file("txt/emp_c2.txt").import@t()		/加载emp_c2数据
3	=A1|A2	0	/合并数据，得到排列
4	for A3	>B3=B3+A4.SALARY	/累计工资。
5	>output("Total=" / B3)		/输出工资总额

输出：Total=23200

10.5.3 排列与序表的不同

比较序表、排列进行**交集、并集**运算。

下面两个排列数据均来自同一序表，再将排列数据分别存入两个新序表，然后进行交集"^" 、并集"&"运算。

	A	B	C
1	=file("txt/user_1.txt").import@t()		/导入数据
2	=A1([1, 2, 3])	=A1([2, 3, 4])	/生成排列
3	=A1.create()	>A3.insert@r(0:A2)	/生成序表，数据来自A2
4	=A1.create()	>A4.insert@r(0:B2)	/生成序表，数据来自B2
5	=A3^A4		/序表间交集，总返回空
6	=A3&A4		/序表间并集，相当于和集
7	=A2^B2		/排列间交集，返回共同记录
8	=A2&B2		/排列间并集，返回并集

A2、A3数据：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11

B2、A4数据：

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	15
2	3	July	F	11
3	4	Gail	M	11

10.5.3 排列与序表的不同

序表集合运算：

A3^A4数据：

Index	Member

不同序表**交集**为空列；

A3&A4数据：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	2	Tom	M	15
5	3	July	F	11
6	4	Gail	M	11

序表**并集**为两表所有的成员，结果等同于和集。

这表明不同序表的记录总是不同的，与字段、字段值是否相同无关。

排列集合运算：

A2^B2数据：

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	15
2	3	July	F	11

排列**交集**为两个集合的共同成员；

A2&B2数据：

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

排列**并集**为两个集合去掉重复数据后的成员。

排列的集合运算**注意事项**：

1、排列具有相同的结构。

2、排列数据来自同一序表。

10.6 习题

10.6 习 题

【习题10.1】计算购物花费的金额。每件商品金额为商品单价乘以商品数量，然后累计所有的商品金额即为消费总额。下面为购物清单：

Index	ID	NAME	PLACE	UNIT	UNITPRICE	NUM
1	1	Apple	MI	1	1.69	7.0
2	2	Banana	FL	1	0.69	6.5
3	3	Cucumber	CA	3	0.77	4.5
4	4	Onion	CA	1	0.99	4.0
5	5	Orange	FL	2	4.99	10.0

	A	B	C
1	=file("foods2.txt").import@t()	0	/B1商品金额变量
2	for A1	>B1=B1+A2.UNITPRICE*A2.NUM	/累计商品金额
3	>output("pay = " / B1)		/输出消费总额

输出：pay = 73.64

A2执行for循环每一件商品。在循环中，B2计算商品单价乘以商品数量即商品金额，累加到商品金额。直到循环结束，累计商品金额即为消费总额。

10.6 习 题

【习题10.2】删除表中GENDER="M"的记录。

	A	B	C	D
1	=file("txt/user_1.txt").import@t()	=A1.count()		/序表记录数
2	for B1,1	if A1(A2).GENDER=="M"	>A1.delete(A2)	/删除记录

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

删除后

Index	SID	NAME	GENDER	AGE
1	3	July	F	11

for循环遍历找出GENDER= “M” 的记录，然后将此记录删除。考虑到删除影响表数据次序，采用了倒序删除。

10.6 习 题

【习题10.3】找出最小年龄用户。

	A	B	C	D
1	=file("txt/employee.txt").import@t()	=A1(1).BIRTHDAY		/C1存储年龄最小记录
2	for A1	=A2.BIRTHDAY		/计算年龄
3		if (B1<B2)	>C1=A2	/存储年龄较小的记录
4			>B1=B2	/存储年龄较小的值
5	>output("user="/C1.NAME/";AGE ="/ age(C1.BIRTHDAY))			

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	…
1	1	Rebecca	Moore	F	California	1974-11-20	…
2	2	Ashley	Wilson	F	New York	1980-07-19	…
3	3	Rachel	Johnson	F	New Mexico	1970-12-17	…
…	…	…	…				

输出：user=Emily; AGE=32

循环时通过比较出生日期，找到较小年龄用户的同时还要存储对应的记录，以便循环结束后知道是哪个用户年龄最小。本题如果存在多个同岁的最小年龄用户，只返回第1个。

10.6 习 题

【习题10.4】插入记录的T.insert@r(k:A)与T.insert@f(k:A) 不同。

	A	B	C
1	=file("txt/user_5.txt").import@t()		
2	=A1.create()	=create(NAME,SID,AGE,GENDER)	=create(NAME,SID,AGE,DEPT)
3	>A2.insert@r(0: A1)	>B2.insert@f(0: A1)	>C2.insert@f(0: A1)
4	/同结构之间数据复制	/同字段名不要求字段顺序相同	/同字段名不要求字段个数相同

表A2数据：

Index	SID	NAME	GENDER	AGE
1	10	Jack	M	
2	11	Tom	M	
3	12	July	F	

表B2数据：

Index	NAME	SID	AGE	GEND
1	Jack	10	12	M
2	Tom	11	15	M
3	July	12	11	F

表C2数据：

Index	NAME	SID	AGE	DEPT
1	Jack	10	12	(null)
2	Tom	11	15	(null)
3	July	12	11	(null)

插入有相同字段而不同字段顺序的记录时，用T.insert@f(k:A)比较方便，它自动将A的记录数据按同名字段赋给T的新增记录；而T.insert@r(k:A) 则按字段顺序赋值给T的新增记录。

10.6 习 题

【习题10.5】对不同条件的记录作相应的修改。

下面根据不同的UNIT字段值，修改商品折扣后的价格：

	A	B	C
1	=file("foods.txt").import@t()		/加载文本文件数据
2	for A1	if A2.UNIT==1	>A2.UNITPRICE=round(A2.UNITPRICE*(1-0.30),2)
3		else if A2.UNIT==2	>A2.UNITPRICE=round(A2.UNITPRICE*(1-0.25),2)
4		else	>A2.UNITPRICE=round(A2.UNITPRICE*(1-0.20),2)

Index	ID	NAME	PLACE	UNIT	UNITPRICE	SELLERS
1	1	Apple	MI	1	1.69	Ashley Williams
2	2	Banana	FL	1	0.69	Andrew Miller
3	3	Cucumber	CA	3	0.77	Christina Williams
4	4	Onion	CA	1		
5	5	Orange	FL	2		

价格修改

Index	ID	NAME	PLACE	UNIT	UNITPRICE	SELLERS
1	1	Apple	MI	1	1.18	Ashley Williams
2	2	Banana	FL	1	0.48	Andrew Miller
3	3	Cucumber	CA	3	0.62	Christina Williams
4	4	Onion	CA	1	0.69	Christina Williams
5	5	Orange	FL	2	3.74	Ashley Williams

多个不同的条件在循环体中作相应的处理，可实现比较复杂的计算逻辑。

10.6 习 题

【习题10.6】计算表中Score字段中的数据总和。

Score数据类型为字符串，求总和时需将字符串按逗号拆分，然后再计算累计各值

Index	DeviceID	Month	Score
1	1050	1	73,85
2	1020	2	66,73,102,140
3	1012	3	81,72,63,68

	A	B	C	D
1	=file("myscore.txt").import@tq()	0		/B1用于存储数据值
2	for A1	=A2.Score.split@cp()		/将字符串拆分成数组
3		for B2	>B1=B1+B3	/累计数据值
4	>output("total="/B1)			/输出数据之和

输出：total=823

针对Score字段中的数据求和，先将每条记录的数字串拆分成整数序列，B3遍历序列B2的每个成员，C3累计这些数据。

10.6 习 题

【习题10.7】某字段多行合并问题。有一鉴定明细单的“原因分析”字段为3行组成(原因现象、诊断结果、维修方案与效果)，需要将其合并，数据如下：

Details of identification sheet

No	Num	Cause analysis	Code	Flag	Assembly plant	Price	Pdate
2408-B190094	1	Fault phenomenon: A1◦ Diagnostic results: B1◦ Maintenance plan and effect: C1◦	E123	12JS160T-1707140 FAST 18H20 009C 18H170209	E123	849.23	2020-4-10
1746-B200001	1	Fault phenomenon: A2◦	E123	FAST GEAR 18A161BXY	0KF	1494.55	2020-3-18
		Diagnostic results: B2◦					
		Maintenance plan and effect: C2◦					
2030-B200049	1	Fault phenomenon: A3◦	fst	FAST 19F02	E123	1962.09	2020-4-3
		Diagnostic results: B3◦					
		Maintenance plan and effect: C3◦					
2137-B200028	1	Fault phenomenon: A4◦	E123	JS130T-1701180 FAST GEAR 18L07	E123	916.24	2020-4-7
		Diagnostic results: B4◦					
		Maintenance plan and effect: C4◦					
2408-B190148	1	Fault phenomenon: A5◦	E123	12JSD160T-1707140-2 FAST 17C30	E123	1899.49	2020-3-31
		Diagnostic results: B5◦					
		Maintenance plan and effect: C5◦					

分析：处理结果是将字段“原因分析”合并到紧邻的记录中。因此遍历数据时，若记录的字段No不为空，则此记录r插入表中，若为空，则将“原因分析”与上次记录r的“原因分析”字段合并。

10.6 习 题

【习题10.7】某字段多行合并问题。

	A	B	C	D
1	=file("ident.xlsx").xlsimport@t(;,2)	=A1.create()		/导入数据
2	for A1	if A2.No==null		/判断是否为空
3			>C1.'Cause analysis'=C1.'Cause analysis'+"; "+A2.'Cause analysis'	/将记录值拼接成字符串并修改记录
4		else		
5			>B1.insert@r(0:[A2])	/向表B1追加数据
6			=C1=B1(B1.len())	/将新增加的记录存到C1
7	=file("ident.xlsx").xlsexport@t(B1;"merge")			/将数据输出到excel文件

A1：f .xlsimport@t(;,2)中的2表示从第二行开始读。

A7：f .xlsexport@t(B1; "merge")表示将B1中的数据存储到名称为“ merge ”的表单中

	A	B	C	D	E	F	G	H
1	No	Num	Cause analysis	Code	Flag	Assembly p	Price	Pdate
2	2408-B19C	1	Fault phenomenon: A1。 Diagnostic results： B1。 Maint	E123	12JS160T-	E123	849.23	2020-04-10
3	1746-B20C	1	Fault phenomenon: A2。 ; Diagnostic results： B2。 ; Mai	E123	FAST GEAR	0KF	1494.55	2020-03-18
4	2030-B20C	1	Fault phenomenon: A3。 ; Diagnostic results： B3。 ; Mai	fst	FAST 19FC	E123	1962.09	2020-04-03
5	2137-B20C	1	Fault phenomenon: A4。 ; Diagnostic results： B4。 ; Mai	E123	JS130T-17	E123	916.24	2020-04-07
6								

10.6 习 题

【习题10.8】多个同结构的数据文件合并。在users目录下有user_#i.txt形式的多个同结构数据文件进行合并(#i为序号)。

	A	B	C	D
1	[]			/定义空列
2	for 10	="users/user_"+string(A2)+".txt"		/拼接成users/user_#i.txt字符串
3		if file(B2).exists()	=file(B2).import@t()	/若文件存在则加载其数据
4			>A1=A1|C3	/数据合并到A1
5	>file("users.txt").export@t(A1)			/将表数据输出到文件users.txt中

B3：判断文件是否存在，C3：加载文件数据，C4：合并文件数据。

Index	SID	NAME	GENDER	AGE
1	10	Jack10	M	19
2	11	Lucy11	F	30
3	12	Tom12	M	25

Index	SID	NAME	GENDER	AGE
1	20	Jack20	M	29
2	21	Lucy21	F	30
3	22	Tom22	M	25

合并数据

Index	SID	NAME	GENDER	AGE
1	10	Jack10	M	19
2	11	Lucy11	F	30
3	12	Tom12	M	25
4	20	Jack20	M	29
…	…	…	…	…

第11章 有关表的计算

11.1 聚合

11.2 过滤与定位

- 过滤
- 定位
- 定位计算

11.3 对位计算

- 修改
- 产生
- 附加

11.4 排序与排名

- 排序
- 排序定位
- 排名

11.5 极值问题

- 极值
- TOP

11.6 习题

11.1 聚合

11.1 聚合

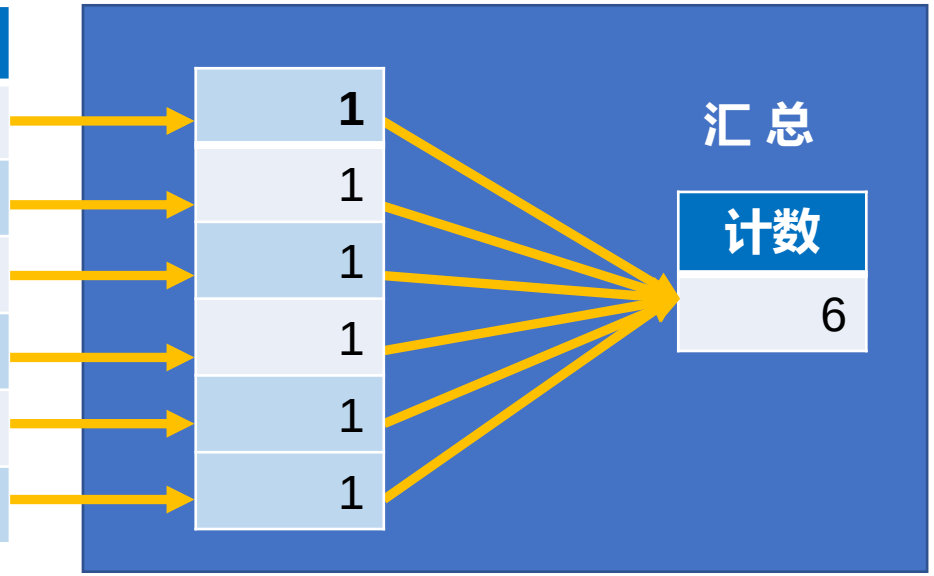
我们在上一章用循环语句实现了聚合运算，这样当然没问题，但比较麻烦。

序表和序列类似，可以作为一个集合整体来运算，使用函数实现聚合。如同在第9章中介绍的序列的运算，对序表或者排列也可以用聚合函数执行聚合运算，如**T.count(x)，T.sum(x)，T.avg(x)，T.min(x)，T.max(x)**等。

需要注意的是，对序表或者排列使用聚合函数时，聚合表达式**x**往往是针对**某个字段或某些字段**的值执行聚合计算的。

计数：count()对表记录计数。

Index	SID	NAME
1	1	A1
2	2	A2
3	3	A3
4	4	A4
5	(null)	(null)
6	5	A5



count()对表记录计数时，各个字段均为空值的**空记录**也要计数。

11.1 聚合

【例11.1】**计数**，求用户表中有多少条记录？数据从文件user_101.txt读取。

解题思路：
A.count()计算表中记录的个数，包括空记录。

	A	B
1	=file("txt/user_101.txt").import@t()	/导入数据
2	=A1.count()	/计数记录数

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11
5	(null)	(null)	(null)	(null)
6	(null)	(null)	(null)	(null)
7	7	Liky	F	10

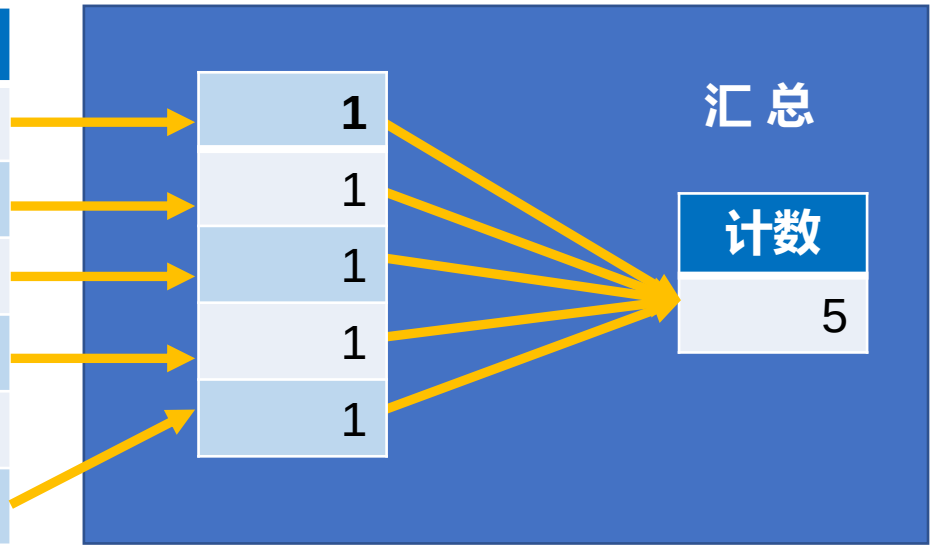
A2	Value
	7

共有7条记录，其中包括两条空记录。

A.count()计数包括空记录。

计数：count(x)对字段值计数。

Index	SID	NAME
1	1	A1
2	2	A2
3	3	A3
4	4	A4
5	(null)	(null)
6	5	A5



count(NAME)对NAME值计数时，空字段值忽略不计，不包括在内。

11.1 聚合

【例11. 2】**计数**，求用户表中有多少个用户？数据从文件user_101.txt读取。

解题思路：
查看非空用户名个数，**A.count(x)**计算表达式**x**结果非空的记录数。

	A	B
1	=file("txt/user_101.txt").import@t()	/导入数据
2	=A1.count(NAME)	/计数用户数

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11
5	(null)	(null)	(null)	(null)
6	(null)	(null)	(null)	(null)
7	7	Liky	F	10

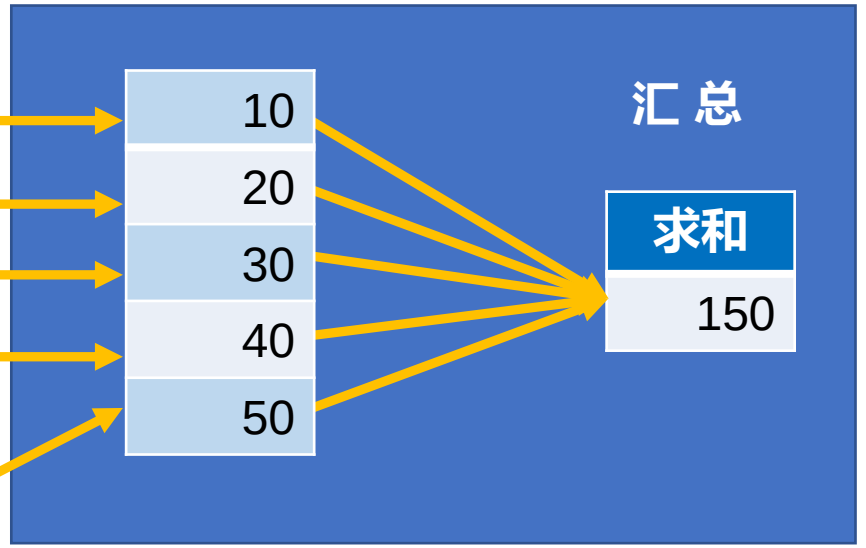
A2	Value
	5

表记录数为7条；用户个数为5，不包括2个空值

A.count(x)计算表达式x结果非空的记录数。

求和：sum(x)对字段值累计相加求和。

Index	SID	SCORE
1	1	10
2	2	20
3	3	30
4	4	40
5	(null)	(null)
6	5	50



sum(SCORE)对的成绩求和时，空值忽略不计，不包括在内。

11.1 聚合

【例11. 3】**求和**，求所有用户年龄之和。数据从文件user_101.txt读取。

解题思路：
累计记录中的用户年龄值，**A.sum(x)**计算表A中x数值列的数据之和。

	A	B
1	=file("txt/user_101.txt").import@t()	/导入数据
2	=A1.sum(AGE)	/计算用户年龄并累计

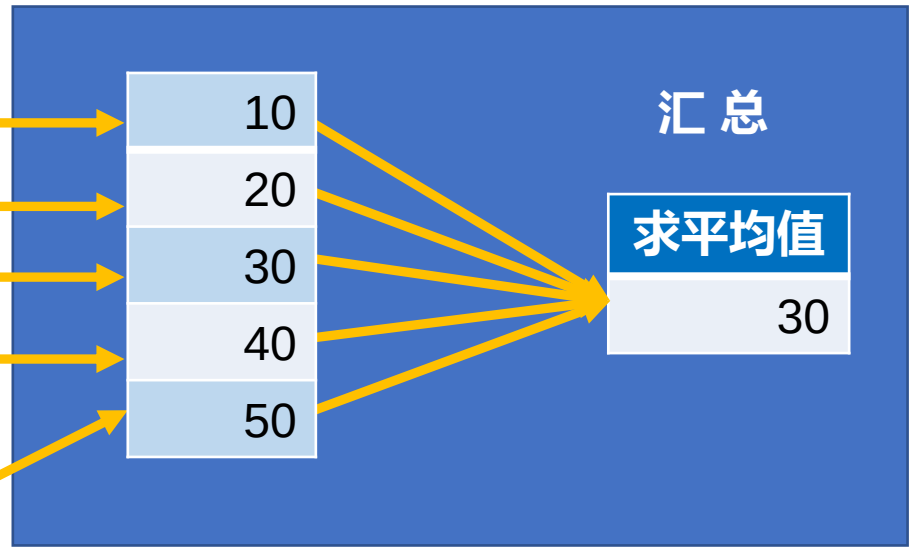
Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11
5	(null)	(null)	(null)	(null)
6	(null)	(null)	(null)	(null)
7	7	Liky	F	10

A2	Value
	49

A.sum(x)对空值忽略不计。

求平均值：avg(x)对字段值计算平均值。

Index	SID	SCORE
1	1	10
2	2	20
3	3	30
4	4	40
5	(null)	(null)
6	5	50



avg(SCORE)对成绩求平均值时，空字段值不包括在内。

11.1 聚合

【例11. 4】**求平均值**，求用户的平均年龄。数据从文件user_101.txt读取。

解题思路：
A.avg(x)计算表A中数值列x数据的平均值。

	A	B
1	=file("txt/user_101.txt").import@t()	/导入数据表
2	=A1.avg(AGE)	/计算平均年龄

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11
5	(null)	(null)	(null)	(null)
6	(null)	(null)	(null)	(null)
7	7	Liky	F	10

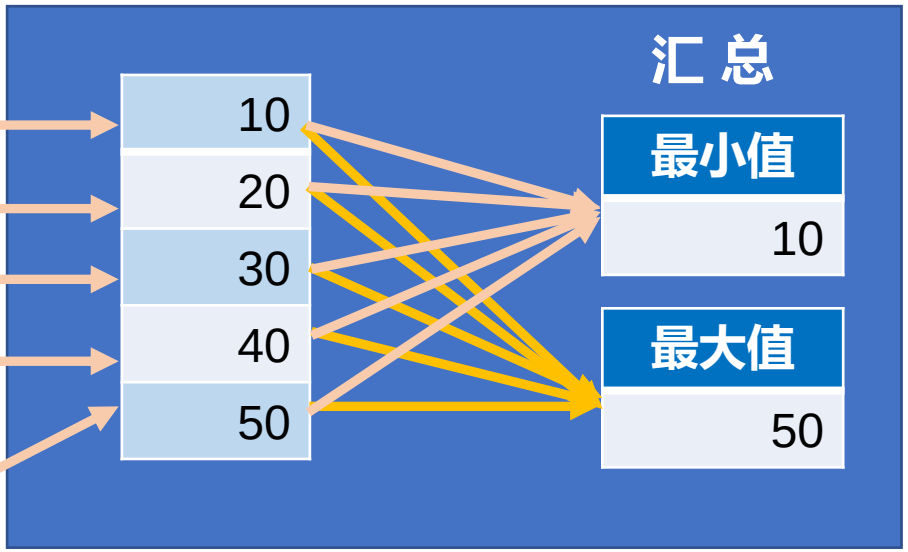
A2	Value
	11.8

输出平均值：**T.avg()**
计算用户年龄平均年龄，忽略空记录的影响。

11.1 聚合

求极值： 对字段值求最大值，对字段值求最小值。

Index	SID	SCORE
1	1	10
2	2	20
3	3	30
4	4	40
5	(null)	(null)
6	5	50



对SCORE成绩求最大值、最小值时，空字段值忽略不计。

11.1 聚合

【例11. 5】**求极值**， 找出用户年龄最大(小) 值。数据从文件user_101.txt读取。

解题思路：
A.max(x)求表A列x中数据的最大值；同理，**A.min(x)**求最小值 。

	A	B
1	=file("txt/user_101.txt").import@t()	/读入文本文件数据
2	=A1.max(AGE)	/计算用户年龄最大值
3	=A1.min(AGE)	/计算用户年龄最小值

A2	Value
	15

A3	Value
	10

11.2 过滤与定位

11.2.1 过滤

过滤使用**关系运算符**>，>=，==，<=，<等。

【例11.6】查询NAME为Ashley的员工。数据从文件emp.txt读取。

解题思路：
对表过滤可用**A.select(x)**，表达式**x**的值不是**false**或**null**则符合过滤条件。**@1**选项表示返回满足条件的第1条记录；忽略**x**则返回所有记录。A.select(x)在序表或排列中的使用，和序列中的使用是相同的。

	A	B
1	=file("txt/emp.txt").import@t()	/加载文本数据，返回序表
2	=A1.select(NAME=="Ashley")	/查询NAME==Ashley的记录
3	=A1.select@1(NAME=="Ashley")	/查询首条记录

A2	Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
	1	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000
	2	5	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000
	3	68	Ashley	Moore	F	New York	1980-05-09	2007-04-01	Sales	10000
	4	80	Ashley	Johnson	F	Illinois	1974-10-20	2004-09-01	Marketing	7000
	…	…	…	…	…	…	…	…	…	…

A3	Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
	1	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000

需要注意的是，A2中得到的结果是记录构成的**排列**，而并不是新序表。

11.2.1 过滤

过滤使用**逻辑运算符**&&、||、！等。

【例11.7】查询NAME为Ashley且SURNAME为Simith的员工。数据从文件emp.txt读取。

解题思路：

A.select(x) 支持多个查询条件。

	A	B
1	=file("txt/emp.txt").import@t()	/加载文本数据，返回序表
2	=A1.select(NAME=="Ashley" && SURNAME== "Smith")	/多条件查询员工

A2

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	5	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000
2	123	Ashley	Smith	F	Indiana	1987-02-09	2011-03-01	Sales	5000
3	163	Ashley	Smith	F	Kentucky	1979-12-26	2010-12-01	HR	5000
4	339	Ashley	Smith	F	Missouri	1980-04-17	2001-11-01	Production	7000
5	376	Ashley	Smith	F	Michigan	1969-08-07	2000-07-01	Marketing	7000

11.2.1 过滤

根据需要，还可以使用其它的**表达式**作为过滤条件，如字符串匹配查询。

【例11.8】查询州名以New开头且工龄大于10年的员工。数据从文件emp.txt读取。

解题思路：
通过字符串匹配查找New开头的州名，工龄可通过age函数计算

	A	B
1	=file("txt/emp.txt").import@t()	/加载文本数据，返回序表
2	=A1.select(left(STATE, 3)=="New" && age(HIREDATE)>10)	/查询州名以New开头且工龄大于10年的员工

A2

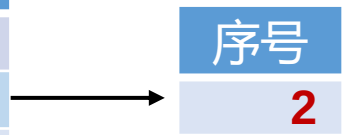
Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000
2	12	Jessica	Davis	F	New York	1980-09-11	2008-09-11	Sales	7000
3	15	Alexis	Smith	F	New York	1983-07-10	2006-07-10	Sales	8000
4	25	Sarah	Davis	F	New York	1976-11-25	2006-11-25	Marketing	12000
5	38	Matthew	Johnson	M	New York	1972-11-20	2002-11-20	R&D	6000
…	…	…	…	…	…	…	…	…	…

11.2.2 定位

有时，我们不只是关心满足条件的记录，而是更关注满足条件的第一条记录所在的位置。
与第9章中介绍过的在序列中定位成员类似，可以用序列的定位函数在序表或者排列中定位记录。

在数据表中查找AVAILABLE=1的首记录序号。

Index	SID	AVAILABLE
1	101	0
2	102	1
3	103	1
4	104	0
5	105	0
6	106	1
7	107	1
8	108	0



序号
2

返回记录序号

当找到满足条件的记录就中止查找，立即返回对应的**序号**。
如我们向自动存储柜放东西时，只需要返回一个空柜就行。

11.2.2 定位

【例11.9】查出比Sydney更早入职的员工 。数据从文件emp.txt读取。

解题思路：
在有序表中定位到指定员工的位置，则此位置前的记录均为要找到的记录， **A.pselect(x)**查找第一个满足查找条件x的记录的序号。

	A	B
1	=file("txt/emp3.txt").import@t().sort(HIREDATE)	/加载数据，按入职时间排序
2	=A1.pselect(NAME:"Sydney")	/返回满足条件的第一条记录序号
3	=A1.to(A2-1)	/通过序号找出员工记录

A3

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	18	Jonathan	Moore	M	Florida	1971-03-07	2000-03-07	Administration	7000
2	203	Abigail	Smith	F	Arizona	1972-02-20	2000-04-01	Sales	7000
3	487	Hannah	Brown	F	Pennsylvania	1980-08-27	2000-05-01	Marketing	5000
4	425	Joseph	Thomas	M	Illinois	1971-07-02	2000-05-01	Sales	7000
5	226	William	Smith	M	New York	1976-01-23	2000-05-01	Sales	6500
…	…	…	…	…	…	…	…	…	…

11.2.2 定位

查询数据时，找出所有**满足**条件的**记录序号**。

在数据表中查找所有AVAILABLE=1的记录序号。

Index	SID	AVAILABLE
1	101	0
2	102	1
3	103	1
4	104	0
5	105	0
6	106	1
7	107	1
8	108	0

序号
2
3
6
7

返回序号
组成的序列

找到所有满足条件的记录序号，返回由序号组成的序列。
如了解自动存储柜哪些是空柜。

11.2.2 定位

【例11.10】找出股票000062收盘价高于10那些天中最大的涨跌额是多少？数据从文件stockrecords.txt读取。

解题思路：
A.pselect()可查找收盘价大于10的记录，再从过滤后的记录中用A.max(x)获取涨跌额最大的记录。使用A.pselect@a(x)找出所有满足条件的记录序号所组成的序列，这和第9章中序列的定位函数使用是相同的。

	A	B
1	=file("txt/stockrecords.txt").import@tq().select(STOCKID=="000062") .sort(DATE)	/加载数据
2	=A1.pselect@a(CLOSING>10)	/返回所有收盘价大于10的记录序号
3	=A2.max(if (~>1, A1(~).CLOSING-A1(~-1).CLOSING, 0))	/返回最大涨跌额

A2

Index	Member
1	34
2	76
3	77
4	79
5	80
…	…

A3

Value
1.1600000000000001

A.pselect@a(x) 返回记录序号序列，A.max(x)获取相邻记录的最大差值。

11.2.2 定位

在**有序**数据表中查询数据时，若找不到则返回满足条件的数据插入位置的相反数。

在按SCORE**排序**的数据表中，查找SCORE=100的记录。

Index	SID	SCORE
1	101	50
2	102	84
3	103	90
4	104	143
5	105	147
6	106	150
7	107	154
8	108	183

返回记录序号

由结果判断记录是否存在，若不存在则可将数据插入有序数据表中。
注意：此种情况要求记录有序，否则返回的结果有误。

11.2.2 定位

【例11.11】将编号BK_1010的新书BOOK110存入有序数据表中。数据从文件books.txt读取。

解题思路：
在按BOOKID排序的图书表中查找编号为BK_1010的书，若找不到则返回值是BK_1010可插入位置的相反数，可通过**A.pselect@s(x)**实现，然后根据返回序号将数据插入。

	A	B	C
1	=file("txt/books.txt").import@t()		/导入数据
2	=A1.pselect@s(BOOKID:"BK_1010")		/查询新书位置
3	if (A2<0)	>A1.insert(-A2, "BK_1010", "BOOK110")	/将书记录插入序表

A1

Index	BOOKID	NAME
1	BK_1001	BOOK101
2	BK_1002	BOOK122
3	BK_1003	BOOK103
4	BK_1005	BOOK114
5	BK_1008	BOOK115
6	BK_1011	BOOK136

A1

Index	BOOKID	NAME
1	BK_1001	BOOK101
2	BK_1002	BOOK122
3	BK_1003	BOOK103
4	BK_1005	BOOK114
5	BK_1008	BOOK115
6	BK_1010	BOOK110
7	BK_1011	BOOK136

A2

Value
-6

A.pselect(x)返回值为-6，其值相反数正是要插入的位置，序表还是保持有序。

11.2.3 定位计算

有了记录序号后，可通过序号进行有关数据计算。

【例11.12】找出股票000062收盘价最高那天的涨跌额是多少？数据从文件stockrecords.txt读取。

解题思路：

和序列中的使用类似，可以用A.pselect()可查找股价最高的记录位置，再用A.calc()计算股票的涨跌额。

A.calc(k, x)在指定序号的记录处计算表达式。**k**为记录序号，**x**为计算表达式，可使用**~**引用当前记录，**~.F** 或 **F** 引用当前记录的 **F** 字段的值，**~.F[-i]** 或 **F[-i]** 相对引用前面第 **i** 条记录的 **F** 字段的值。

	A	B
1	=file("txt/stockrecords.txt").import@tq().select(STOCKID=="000062"). sort(DATE)	=A1.max(CLOSING)
2	=A1.pselect(CLOSING==B1)	/返回股价最高的记录序号
3	=A1.calc(A2, CLOSING - CLOSING[-1])	/在指定序号的记录处计算涨跌额

A3	Value
	1.1300000000000008

11.2.3 定位计算

有了由记录序号组成的序列后，也可通过此序列进行有关数据计算。

【例11.12】找出股票000062收盘价最高那天的涨跌额是多少？数据从文件stockrecords.txt读取。

解题思路：

A.pselect()可查找股价大于10的记录，再从过滤后的记录中用A.calc()计算股票的涨跌额。

A.calc(p, x) p为记录序号组成的序列，x为计算表达式，对指定的记录计算表达式，返回计算结果。

	A	B
1	=file("txt/stockrecords.txt").import@tq().select(STOCKID=="000062")	/加载数据
2	=A1.pselect@a(CLOSING>10)	/返回所有股价大于10的记录序号
3	=A1.calc(A2, CLOSING - CLOSING[-1])	/通过记录序号计算涨跌额

A3	Index	Member
	1	0.1899999999999995
	2	0.8400000000000016
	3	0.1999999999999993
	4	0.8599999999999994
	5	0.9100000000000001
	…	…

A.calc()通过不同类型的参数，能对单个记录或多个记录进行定位计算。

11.3 对位计算

11.3.1 修改

需要将字段GENDER中的M改为Male, F改为Female， SPL脚本for循环方式实现如下：

Index	SID	NAME	GENDER	BIRTHDAY
1	1	Jack	M	2007-05-03
2	2	Tom	M	2004-07-12
3	3	July	F	2010-09-04
4	4	Gail	M	2010-03-08

Index	SID	NAME	GENDER	BIRTHDAY
1	1	Jack	Male	2007-05-03
2	2	Tom	Male	2004-07-12
3	3	July	Female	2010-09-04
4	4	Gail	Male	2010-03-08

	A	B	C
1	=file("txt/user_2.txt").import@t()		/导入数据
2	for A1	if A2.GENDER=="M"	>A2.GENDER="Male"
3		else	>A2.GENDER="Female"

修改某列字段值，for循环方式需要多行代码实现，而使用函数A.run(x)书写就简便多了。

11.3.1 修改

【例11.14】将字段GENDER中的M改为Male, F改为Femal。数据从文件user_2.txt读取。

解题思路：
针对表中的每条记录修改，可通过**A.run(x)**函数实现。
A.run(x) 对表的每条记录计算表达式 **x**，用计算结果修改当前记录，最后返回修改后的 **A**。

	A	B
1	=file("txt/user_2.txt").import@t()	/导入数据
2	>A1.run(GENDER=if (GENDER=="M", "Male" , "Female"))	/将性别"F"，"M"改为对应的"Female"，"Male"

A1

Index	SID	NAME	GENDER	BIRTHDAY
1	1	Jack	M	2007-05-03
2	2	Tom	M	2004-07-12
3	3	July	F	2010-09-04
4	4	Gail	M	2010-03-08

A1

Index	SID	NAME	GENDER	BIRTHDAY
1	1	Jack	Male	2007-05-03
2	2	Tom	Male	2004-07-12
3	3	July	Female	2010-09-04
4	4	Gail	Male	2010-03-08

注：**A.run(x)**中的表达式x是**赋值类的表达式。**

11.3.1 修改

对多字段值修改， 同时对多个字段的值进行修改。

【例11.15】对用户表信息的性别修改、出生日期转换成年龄。数据从文件user_2.txt读取。

解题思路：

表记录中有多个字段值需要修改时，即计算多个表达式，可通过**A.run(x_1,x_2, ...x_i)**函数实现。

类似A.run(x)，**A.run(x_1,x_2, ...x_i)**针对的是多个表达式xi，可使用"~"引用当前记录。

	A	B
1	=file("txt/user_2.txt").import@t()	/导入文件数据
2	>A1.run(GENDER=if (GENDER=="M", "Male" , "Female"),BIRTHDAY=age(BIRTHDAY))	/计算多个表达式

A1

Index	SID	NAME	GENDER	BIRTHDAY
1	1	Jack	M	2007-05-03
2	2	Tom	M	2004-07-12
3	3	July	F	2010-09-04
4	4	Gail	M	2010-03-08

A2修改

A1

Index	SID	NAME	GENDER	BIRTHDAY
1	1	Jack	Male	13
2	2	Tom	Male	16
3	3	July	Female	9
4	4	Gail	Male	10

11.3.2 产生

需要获取指定字段的数据时，如：

Index	SID	NAME	GENDER	BIRTHDAY
1	1	Jack	M	2007-05-03
2	2	Tom	M	2004-07-12
3	3	July	F	2010-09-04
4	4	Gail	M	2010-03-08

Index	Member
1	2007-05-03
2	2004-07-12
3	2010-09-04
4	2010-03-08

访问BIRTHDAY字段数据或通过出生日期计算年龄时，则for循环方法如下：

	A	B	C
1	=file("txt/user_2.txt").import@t()		/A2用来存储BIRTHDAY值
2	[]	[]	/B2用来年龄值
3	for A1	>A2=A2|A3. BIRTHDAY	/循环存储BIRTHDAY值
4	for A1	>B2=B2|age(A4. BIRTHDAY)	/循环存储年龄值

无论是直接取字段值，还是对字段计算后获取其值，还有比for循环方式更简单的方法。

11.3.2 产生

【例11.16】获取表中出生日期字段的数据，并计算用户年龄。从数据文件user_2.txt读取。

解题思路：
获取表列数据时，如出生日期字段数据，可通过**A.(x)** 获取，**A.(x)**的使用和第9章中在序列中时相同。
在序表或排列中使用时，**A.(x)** 对表中每条记录计算表达式**x**，返回计算结果值组成的序列。**x为空**时，返回表**A所有的数据；x**一般为字段名，或者由字段名组成的表达式，在表达式中，可使用**~**引用当前记录。

	A	B
1	=file("txt/user_2.txt").import@t()	/导入数据
2	=A1.(BIRTHDAY)	/获取出生日期的数据
3	=A1.(age(BIRTHDAY))	/通过出生日期计算年龄

A1

Index	SID	NAME	GENDER	BIRTHDAY
1	1	Jack	M	2007-05-03
2	2	Tom	M	2004-07-12
3	3	July	F	2010-09-04
4	4	Gail	M	2010-03-08

A2

Index	Member
1	2007-05-03
2	2004-07-12
3	2010-09-04
4	2010-03-08

A3

Index	Member
1	13
2	16
3	9
4	10

11.3.2 产生

根据表中的数据构建新序表。如提取表中NAME， BIRTHDAY字段数据。

原数据表

Index	SID	NAME	GENDER	BIRTHDAY
1	1	Jack	M	2007-05-03
2	2	Tom	M	2004-07-12
3	3	July	F	2010-09-04
4	4	Gail	M	2010-03-08

新数据表

Index	NAME	BIRTHDAY
1	Jack	2007-05-03
2	Tom	2004-07-12
3	July	2010-09-04
4	Gail	2010-03-08

11.3.2 产生

【例11.17】从员工表提取EID、NAME 、SURNAME 、DEPT等字段的数据。数据从文件emp.txt读取。

解题思路：
用表中多个指定字段构造新序表，可通过**A.new(x_i:F_i, ...)**函数实现。
A.new(x_i:F_i, ...)生成与表A记录数相同的新序表，x_i为表达式，计算的结果为字段值；F_i为字段名，省略时则用x_i中解析出的标识符。

	A	B
1	=file("txt/emp.txt").import@t()	/加载文本数据，返回序表
2	=A1.new(EID, NAME, SURNAME, DEPT)	/用EID，NAME等字段生成新表。

A2

Index	EID	NAME	SURNAME	DEPT
1	1	Rebecca	Moore	R&D
2	2	Ashley	Wilson	Finance
3	3	Rachel	Johnson	Sales
4	4	Emily	Smith	HR
5	5	Ashley	Smith	R&D
…	…	…		

A.new(x_i:F_i,...) 若 **F_i** 省略则用 **x_i** 的表达式串做为新字段名

11.3.2 产生

从现有表构造新序表可以使用计算列。

【例11.18】从员工表获取数据构建新表，将NAME, SURNAME合并成FULLNAME，计算用户年龄存储到新增字段AGE，读取#4列(GENDER) 。数据从文件emp.txt读取。

解题思路：

用现有表的字段及计算列构造新序表，可通过**A.new(x_i:F_i, …)**函数实现。

A.new(x_i:F_i, …)xi为表达式，结果为字段值；当参数x_i使用#i时，表示第i列，此时使用原列名。

	A	B
1	=file("txt/emp.txt").import@t()	/加载文本数据，返回序表
2	=A1.new(EID, NAME+" "+SURNAME:FULLNAME, age(BIRTHDAY):AGE, #4)	/用EID，FULLNAME等字段生成新表，#4表示GENDER字段

A2	Index	EID	FULLNAME	AGE	GENDER
	1	1	Rebecca Moore	45	F
	2	2	Ashley Wilson	39	F
	3	3	Rachel Johnson	49	F
	4	4	Emily Smith	35	F
	5	5	Ashley Smith	45	F
	…	…	…	…	…

11.3.2 产生

根据需要，有时将计算放在生成新表之后再进行。

【例11.19】从员工表获取EID，NAME，SURNAME，BIRTHDAY等字段数据后，计算其年龄。数据从文件emp.txt读取。

解题思路：

用**A.new(x_i:F_i, …)**构造新序表的过程中，可增加无值的新字段AGE，在后续代码用来保存年龄，**A.run(x)**根据出生日期计算年龄并存储到AGE中。

	A	B
1	=file("txt/emp.txt").import@t()	/加载文本数据，返回序表
2	=A1.new(EID, NAME+" "+SURNAME:FULLNAME, BIRTHDAY, AGE)	/用EID，FULLNAME等字段生成新表，其中AGE为空字段。
3	>A2.run(AGE=age(BIRTHDAY))	/计算年龄并写入序表。

A2	Index	EID	FULLNAME	BIRTHDAY	AGE
	1	1	Rebecca Moore	1974-11-20	45
	2	2	Ashley Wilson	1980-07-19	39
	3	3	Rachel Johnson	1970-12-17	49
	4	4	Emily Smith	1985-03-07	35
	5	5	Ashley Smith	1975-05-13	45
	…	…	…	…	…

11.3.3 附加

根据需要，可对现有数据表**追加新字段**来存储有关数据。

成绩表中记录了每个学生的各科成绩，还想要了解每个学生成绩之和，且需要进一步计算，可增加字段记录它。如增加Score字段来记录成绩之和。

Index	SID	Name	English	Math	History	Score
1	101	Tom	50	65	73	188
2	102	July	80	98	66	244
3	103	Jack	68	88	81	237
4	104	Mark	86	78	84	248
5	105	Megan	87	94	85	266

11.3.3 附加

【例11.20】查看成绩表上每个学生的各科成绩及总成绩。数据从文件score_2.txt读取。

解题思路：
成绩表中有每个学生的各科成绩，增加新字段存储总成绩，可通过A.derive()函数实现。
A.derive(x_i:F_i,...)在复制表A的基础上增加新字段构建新序表；x_i 为表达式，计算结果为字段值；F_i 为新增字段名，F_i 不能与A中原有字段同名。

	A	B
1	=file("txt/score_2.txt").import@t()	/加载文本数据，返回序表
2	=A1.derive(English+Math+History: Score)	/追加Score字段记录总成绩

A2	Index	SID	Name	English	Math	History	Score
	1	101	Tom	50	65	73	188
	2	102	July	80	98	66	244
	3	103	Jack	68	88	81	237
	4	104	Mark	86	78	84	248
	5	105	Megan	87	94	85	266

11.3.3 附加

根据需要，有时在构建新表时增加新字段，后续再将数据写入新字段。

【例11.21】了解每个学生的各科成绩之和与平均值。数据从文件score_2.txt读取。

解题思路：

可先通过**A. derive()**追加两个字段，**A.run(x)**计算每个学生的总成绩、各科平均成绩后存到这两个字段中。

	A	B
1	=file("txt/score_2.txt").import@t()	/加载文本数据，返回序表
2	=A1.derive(TScore, AvgScore)	/追加TScore, AvgScore字段
3	>A2.run(TScore=English+Math+History, AvgScore=round(TScore/3, 2))	/将总成绩、平均成绩存到新增字段

A2	Index	SID	Name	English	Math	History	TScore	AvgScore
	1	101	Tom	50	65	73	188	62.67
	2	102	July	80	98	66	244	81.33
	3	103	Jack	68	88	81	237	79.0
	4	104	Mark	86	78	84	248	82.67
	5	105	Megan	87	94	85	266	88.67

A3: TScore用来存储每条记录的总成绩；**round()**将成绩平均值保持两位小数。

11.4 排序与排名

11.4.1 排序

排序就是将一组无序的记录调整为有序的记录。

对数据表而言，常用字段进行排序。排序时，先确定使用哪个或哪些字段，进行**升序**或**降序**方式排序。排序方式缺省时一般都是**升序**排序。这和第9章中介绍过的序列排序有一些区别，序表中是将**记录**排序，排序往往是依据某个字段的值，或者某个表达式的计算结果。排序后返回的结果，是将记录调整顺序后得到的**排列**。

序表数据

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

升序：指从小到大排序；
降序：指从大到小排序

NAME排序是字符串按**ASCII码值**(升序)顺序排序。

按NAME**升序**排序后的排列

Index	SID	NAME↑	GENDER	AGE
1	4	Gail	M	11
2	1	Jack	M	12
3	3	July	F	11
4	2	Tom	M	15

按NAME**降序**排序后的排列

Index	SID	NAME↓	GENDER	AGE
1	2	Tom	M	15
2	3	July	F	11
3	1	Jack	M	12
4	4	Gail	M	11

11.4.1 排序

【例11.22】按出生日期排序查看员工记录。数据从文件emp.txt读取。

解题思路：
A.sort(x)支持对表记录按给定的字段进行排序。
x为表达式，对记录按照x进行升序排序；-x表示降序排列；选项**@z**表示逆序。

	A	B
1	=file("txt/emp.txt").import@t()	/加载文本数据，返回序表
2	=A1.sort(BIRTHDAY)	/按出生日期升序排序显示
3	=A1.sort@z(BIRTHDAY)	/按出生日期降序排序显示
4	=A1.sort(-BIRTHDAY)	/同上

11.4.1 排序

【例11.22】按出生日期排序查看员工记录。数据从文件emp.txt读取。

A2　按BIRTHDAY**升序**排序

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY↑	HIREDATE	DEPT	SALARY
1	296	Olivia	Johnson	F	Ohio	1968-11-05	2006-11-01	Production	10000
2	444	Alexis	Johnson	F	New Hampshire	1968-11-12	2010-12-01	Marketing	6500
3	440	Nicholas	Smith	M	Illinois	1968-11-24	2008-07-01	Marketing	8000
4	158	Luis	Johnson	M	Ohio	1968-11-27	2003-02-01	Technology	8000
5	331	Tyler	Williams	M	Florida	1968-12-23	2001-09-01	Sales	6500
…	…	…	…	…	…	…	…	…	…

A3　按BIRTHDAY**降序**排序

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY↓	HIREDATE	DEPT	SALARY
1	472	Hannah	Brown	F	California	1987-12-27	2008-10-01	Sales	10000
2	89	Emily	Johnson	F	Michigan	1987-12-09	2010-02-01	Technology	7000
3	241	Samantha	Davis	F	Georgia	1987-12-04	2007-01-01	Finance	5000
4	437	Zachary	Williams	M	Tennessee	1987-12-01	2009-02-01	Marketing	6500
5	393	Hannah	Johnson	F	Florida	1987-09-06	2006-01-01	Sales	5000
…	…	…	…	…	…	…	…	…	…

11.4.1 排序

多字段排序，排序时先比较第一个字段，若是升序排序则字段值大的排在后面，字段值小的排在前面，降序反之；若字段值相等，则比较第二个字段，第二个字段的排序规则类同于第一个字段，后续字段也依此类推。

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	11
4	4	Gail	M	11

按AGE**升序**、GENDER**升序**排序后的排列

Index	SID	NAME	GENDER↑	AGE↑
1	3	July	F	11
2	4	Gail	M	11
3	1	Jack	M	12
4	2	Tom	M	15

按AGE**升序**、GENDER**降序**排序后的排列

Index	SID	NAME	GENDER↓	AGE↑
1	4	Gail	M	11
2	3	July	F	11
3	1	Jack	M	12
4	2	Tom	M	15

11.4.1 排序

【例11.23】按NAME升序排序、SURNAME降序排序查看员工记录。数据从文件emp.txt读取。

解题思路：

A.sort(x_1,x_2,.....x_i)支持对表**多字段**进行排序，每个字段可独自指定排序方式。

	A	B
1	=file("txt/emp.txt").import@t()	/加载文本数据，返回序表
2	=A1.sort(NAME, -SURNAME)	/按NAME升序，SURNAME降序排序显示

A2 NAME升序SURNAME降序排序后的排列

Index	EID	NAME↑	SURNAME↓	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	180	Abigail	Smith	F	New York	1972-09-19	2007-05-01	HR	5000
2	203	Abigail	Smith	F	Arizona	1972-02-20	2000-04-01	Sales	7000
3	274	Abigail	Smith	F	Illinois	1978-04-05	2003-08-01	Marketing	5000
4	410	Abigail	Smith	F	Tennessee	1984-06-28	2004-04-01	Marketing	10000
5	311	Abigail	Johnson	F	Florida	1983-09-11	2003-08-01	Production	6500
…	…	…	…	…	…	…	…	…	…

11.4.1 排序

排序时除了可指定字段排序，也可根据计算表达式的值进行排序。

【例11.24】按员工出生日期的月、日排序查看。数据从文件emp.txt读取。

解题思路：
可根据出生日期获取月、日，A.sort(x_i,...) x_i 为多个表达式，计算表达式得到月、日，然后再排序。

	A	B
1	=file("txt/emp.txt").import@t()	/加载学习成绩数据
2	=A1.sort(month(BIRTHDAY), day(BIRTHDAY))	/按出生日期的月、天升序排序

A2 按出生日期的月、天升序排序

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY↑	HIREDATE	DEPT	SALARY
1	172	Brandon	Brown	M	Pennsylvania	1980-01-01	2002-01-01	HR	6500
2	334	Matthew	Jones	M	California	1977-01-01	2000-12-01	Sales	8000
3	449	Jessica	Jones	F	California	1986-01-02	2006-05-01	Production	5000
4	207	Jacob	Smith	M	Tennessee	1972-01-03	2010-12-01	Sales	5000
5	294	Madison	Johnson	F	Michigan	1986-01-04	2010-06-01	Production	10000
…	…	…	…	…	…	…	…	…	…

11.4.1 排序

空值排序，按照指定的某字段排序，若字段中有空值，则空值所在的记录可出现在行首或行尾。

序表数据

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	(null)
4	4	Gail	M	11

按AGE**升序**, null排序在行尾的排列

Index	SID	NAME	GENDER	AGE↑
1	4	Gail	M	11
2	1	Jack	M	12
3	2	Tom	M	15
4	3	July	F	(null)

按AGE**升序**, null排序在行首的排列

Index	SID	NAME	GENDER	AGE↑
1	3	July	F	(null)
2	4	Gail	M	11
3	1	Jack	M	12
4	2	Tom	M	15

在升序排序情况下，有空值的记录位置如上所示。
同理，在降序排序情况下，也存在类似情况，在此不再赘述。

11.4.1 排序

【例11.25】按AGE升序排序，查看空值所在的记录分别在前、在后情况。数据从文件user_12.txt读取。

解题思路：
A.sort(x)支持对表字段进行排序，-x表示降序排序，缺省有空值的记录排在前，选项@0有空值的记录排在后。

	A	B
1	=file("txt/user_12.txt").import@t()	/加载文本数据，返回序表
2	=A1.sort(AGE)	/按AGE升序排序显示，有空值的记录在前
3	=A1.sort@0(AGE)	/按AGE升序排序显示，有空值的记录在后

A1

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	(null)
3	3	July	F	14
4	4	Gail	M	11
5	5	Liky	F	10

A2

Index	SID	NAME	GENDER	AGE
1	2	Tom	M	(null)
2	5	Liky	F	10
3	4	Gail	M	11
4	1	Jack	M	12
5	3	July	F	14

A3

Index	SID	NAME	GENDER	AGE
1	5	Liky	F	10
2	4	Gail	M	11
3	1	Jack	M	12
4	3	July	F	14
5	2	Tom	M	(null)

11.4.2 排序定位

获取序列排序后在原排列的序号。下面演示字母序列排序与人员按AGE排序后显示的序号

排序后字母在原序列中的序号

Char	a	c	e	g	f	d	b
序号	1	2	3	4	5	6	7
	排序后序号						
Char	a	b	c	d	e	f	g
序号	1	7	2	6	3	5	4

表数据

Index	SID	NAME	GENDER	AGE
1	1	Jack	M	12
2	2	Tom	M	15
3	3	July	F	13
4	4	Gail	M	11
5	5	Liky	F	10

排序

按AGE排序

Index	SID	NAME	GENDER	AGE↑
1	5	Liky	F	10
2	4	Gail	M	11
3	1	Jack	M	12
4	3	July	F	13
5	2	Tom	M	15

返回序号

原序列Index

原Index
5
4
1
3
2

11.4.2 排序定位

【例11.26】查出来自Ohio州的员工，按DEPT, HIREDATE排序后在原表中的序号。数据从文件emp.txt读取。

解题思路：
获取Ohio州的员工按排序后记录在原序列中的序号，再根据序号生成新序表同时将序号追加到新序表。
A.psort(x)中 x 为表达式，获取排序后记录在原序列中的序号。

	A	B
1	=file("txt/emp.txt").import@t().select(STATE=="Ohio")	/加载文本数据，返回排列
2	=A1.psort(DEPT, HIREDATE)	/按DEPT, HIREDATE同时升序排序获取序号
3	=A1(A2).derive(A2(#): ROW)	/将A2排名追加到序表

A3

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY	ROW
1	219	Emily	Johnson	F	Ohio	1986-10-25	2007-09-01	Finance	6500	6
2	108	William	Brown	M	Ohio	1978-03-25	2000-10-01	Marketing	6500	2
3	494	Emily	Johnson	F	Ohio	1975-08-12	2001-12-01	Marketing	8000	22
4	488	Hannah	Smith	F	Ohio	1973-04-06	2002-06-01	Marketing	8000	20
5	268	Jasmine	Johnson	F	Ohio	1977-01-09	2008-10-01	Marketing	10000	8
…	…	…	…	…	…	…	…	…	…	…

A3：A(p)根据A2中的序号数列生成序表，A.derive()再增加字段ROW记录对应的序号。

11.4.3 排名

排名，即排列名次。现在流行各种排行榜，如学生成绩排行榜，流行歌曲排行榜，汽车销售月排行榜等，通过排行榜，可了解排名情况。如通过汽车销售月排行榜，我们可知道2020年4月，大众宝来轿车销量23731辆，排名第6名等情况。

排名与排序一样，也有**升序**与**降序**之分，通常情况，缺省排名为**升序**。

按AGE**升序**排名序列

Index	SID	NAME	AGE	Rank↑
1	1	Jack	12	3
2	2	Tom	15	4
3	3	July	11	1
4	4	Gail	11	1

按AGE**降序**排名序列

Index	SID	NAME	AGE	Rank↓
1	1	Jack	12	2
2	2	Tom	15	1
3	3	July	11	3
4	4	Gail	11	3

说明：为了演示排名需要，在源序表基础上增加说明Rank字段，用橙色标注，说明数据记录排名次序情况。后面有类似情况，不再单独列出源序表。

11.4.3 排名

【例11.27】查看科目1成绩升序、降序排名情况。数据从文件scores.txt读取。

解题思路：
对成绩进行排名计算，可用A.ranks(x)。**A.ranks(x)** 计算每条记录对应的**x**值的排名，缺省为升序排名；**-x**表示对x降序排名；选项**@z**表示逆序。

	A	B
1	=file("txt/scores.txt").import@t().select(Subject:1)	/查询课目1
2	=A1.ranks(Score)	/返回按成绩排名(由小到大)
3	=A1.ranks@z(Score)	/返回按成绩排名(由大到小)
4	=A1.ranks(-Score)	/同上

A1 成绩数据表

Index	Student	Subject	Score
1	10001	1	71
2	10002	1	87
3	10003	1	88
4	10004	1	53
5	10005	1	91
6	10006	1	92
…	…	…	…

A2 升序排名

Index	Member↑
1	31
2	69
3	70
4	2
5	83
6	84
…	…

A3 降序排名

Index	Member↓
1	69
2	32
3	28
4	99
5	18
6	16
…	…

11.4.3 排名

排名与紧凑排名

正常排名时，若某名次n并列m个，则紧跟着的下一个为第n+m名，如下图左侧所示，用户表按年龄升序排名，July和Gail并列第2名，则紧跟着的Jack排名第4；而紧凑排名时下一个则为第n+1名，如下图右侧所示，Jack名次为第3。

通常情况，不作特别说明的排名都是正常排名，如前面提到的排名。

按AGE升序排名

Index	SID	NAME	AGE	Rank↑
1	1	Jack	12	4
2	2	Tom	15	5
3	3	July	11	2
4	4	Gail	11	2
5	5	Liky	10	1

按AGE升序紧凑排名

Index	SID	NAME	AGE	Rank↑
1	1	Jack	12	3
2	2	Tom	15	4
3	3	July	11	2
4	4	Gail	11	2
5	5	Liky	10	1

按AGE降序排名

Index	SID	NAME	AGE	Rank↓
1	1	Jack	12	2
2	2	Tom	15	1
3	3	July	11	3
4	4	Gail	11	3
5	5	Liky	10	5

按AGE降序紧凑排名

Index	SID	NAME	AGE	Rank↓
1	1	Jack	12	2
2	2	Tom	15	1
3	3	July	11	3
4	4	Gail	11	3
5	5	Liky	10	4

11.4.3 排名

【例11.28】商品销售记录表的升序、降序紧凑排名情况。数据从文件goods.txt读取。

解题思路：
ranks@i (x) 实现对x的紧凑排名，有升序与降序之分。

	A	B
1	=file("txt/goods.txt").import@t()	/查询课目1
2	=A1.ranks@i(SALE)	/升序紧凑排名
3	=A1.ranks@i(-SALE)	/降序紧凑排名

A1 商品销售记录表

Index	SID	PDATE	SALE
1	101	2019-03-01	260
2	102	2019-03-02	184
3	103	2019-03-03	328
4	104	2019-03-04	143
5	105	2019-03-05	210
6	106	2019-03-06	328
7	107	2019-03-07	254
8	108	2019-03-08	239

A2 升序紧凑排名

Index	Member↑
1	6
2	2
3	7
4	1
5	3
6	7
7	5
8	4

A3 降序紧凑排名

Index	Member↓
1	2
2	6
3	1
4	7
5	5
6	1
7	3
8	4

11.4.3 排名

除了排名和紧凑排名外，还有**平均排名**，若正常排名时某名次n并列m个，在**平均排名**下这m个的名次则为n+(m-1)/2。

按AGE升序**排名**

Index	SID	NAME	AGE	Rank↑
1	1	Jack	12	4
2	2	Tom	15	5
3	3	July	11	2
4	4	Gail	11	2
5	5	Liky	10	1

按AGE**平均排名**

Index	SID	NAME	AGE	Rank↑
1	1	Jack	12	4.0
2	2	Tom	15	5.0
3	3	July	11	2.5
4	4	Gail	11	2.5
5	5	Liky	10	1.0

如上图所示，July和Gail在正常排名时并列第2名，而在平均排名时名次则为2+(2-1)/2=2.5。

11.4.3 排名

【例11.29】求对科目1的成绩在平均排名下的名次。数据从文件 scores.txt 中读取。

解题思路：
ranks@s(x) 计算每条记录的x值在平均排名下的名次，可能不是整数。

	A	B
1	=file("txt/scores.txt").import@t().select(Subject:1)	/导入课目1成绩
2	=A1.ranks@s(Score)	/计算平均排名下的名次

A1 科目1成绩的排列

Index	Student	Subject	Score
1	10001	1	71
2	10002	1	87
3	10003	1	88
4	10004	1	53
5	10005	1	91
6	10006	1	92
7	10007	1	69
8	10008	1	76
…	…	…	…

A2 平均排名

Index	Member
1	31.5
2	69.0
3	71.5
4	2.0
5	83.0
6	84.5
7	27.0
8	48.0
…	…

11.4.3 排名

有时我们希望了解某指定的数据在排行榜上的名次。如想了解英文90分，在班级里是什么样的水平。

序表数据

Index	SID	NAME	SCORE	ASCENDING↑	DESCENDING↓
1	1	Jack	80	1	5
2	2	Tom	85	2	3
3	3	July	91	4	2
4	4	Gail	85	2	3
5	5	Liky	95	5	1

90在表中的**升序**排名

Index	Member
1	4

90分在表中的**降序**排名

Index	Member
1	3

90分在表中的**紧凑排名**

Index	Member
1	3

85分在表中的**平均排名**

Index	Member
1	2.5

11.4.3 排名

【例11.30】成绩表中，求科目1比90分多的人数，比90分少的人数？数据从文件 scores.txt 中读取。

解题思路：
A.rank(y, x)计算y值在所有记录的x值中的排名，缺省为升序排名，选项@z为逆序。

	A	B	C
1	=file("txt/scores.txt").import@t().select(Subject:1)		/查询课目1
2	=A1.rank(90, Score)	=A2-1	/返回按成绩排名(由小到大)
3	=A1.rank@z(90, Score)	=A3-1	/返回按成绩排名(由大到小)

A1

Index	Student	Subject	Score
1	10001	1	71
2	10002	1	87
3	10003	1	88
4	10004	1	53
5	10005	1	91
6	10006	1	92
…	…	…	…

比90分少的人数

B2	Value
	79

比90分多的人数

B3	Value
	18

11.4.3 排名

【例11.31】了解每个学生科目1成绩排名情况。数据从文件 scores.txt 中读取。

解题思路：
先构建新表并存入Student, Score的数据，包括升降两个字段，用A.run()从A.ranks() 的结果中得到升序排名、降序排名数据写入升降字段。

	A	B	C
1	=file("txt/scores.txt").import@t().select(Subject:1)	=A1.new(Student, Score, Ascending, Descending)	/B1建立存储排名的序表
2	=A1.ranks(Score)	=A1.ranks@z(Score)	/按Score升序、降序分别排名
3	>B1.run(Ascending=A2(#), Descending=B2(#))		/将排名等信息存入序表

B1	Index	Student	Score	Ascending↑	Descending↓
	1	10001	71	31	69
	2	10002	87	69	32
	3	10003	88	70	28
	4	10004	53	2	99
	5	10005	91	83	18
	6	10006	92	84	16
	…	…	…	…	…

11.4.3 排名

同样，了解每条记录在序表中的排名情况，使用**rank(x, y)**实现。

【例11.32】了解每个学生科目1成绩排名情况。数据从文件scores.txt中读取。

解题思路：

先构建新表并存入Student, Score的数据，包括升降两个字段，用**A.run()**从**A.rank(x, y)**函数中得到升序排名、降序排名数据写入升降字段。

	A	B
1	=file("txt/scores.txt").import@t().select(Subject:1)	/导入科目1成绩数据
2	=A1.new(Student, Score, Ascending, Descending)	/建立存储排名的序表
3	>A2.run(Ascending=A2.rank(~.Score, Score), Descending=A2.rank@z(~.Score, Score))	/计算当前记录的Score在A2的所有Score中的排名

A2	Index	Student	Score	Ascending↑	Descending↓
	1	10001	71	31	69
	2	10002	87	69	32
	3	10003	88	70	28
	4	10004	53	2	99
	5	10005	91	83	18
	…	…	…	…	…

A3中的**~.Score**是指“**A2.run(...)**”中的A2的**当前记录**中的Score值, 后面那个Score则是指“**A2.rank(...)**”中的A2的Score列。

11.5 极值问题

11.5.1 极值

求极值，从数据表中查找最大值或最小值。

如求集合中最大值，会经常遇到下面几种情况：

A、找出最大**值**。

B、找出最大值所在记录的**序号**。

C、找出最大值所在的**记录**。

同理，求最小值也会有类似上面的几种情况。

Index	Student	Score
1	10047	94
2	10004	96
3	10026	72
4	10037	89
5	10073	94
6	10068	71
7	10054	86

Score最高分在上面3种操作下的结果：

A返回值：

值
96

B返回序号：

序号
2

C返回记录：

Student	Score
10004	96

11.5.1 极值

A、找出最大值

【例11.33】查询科目1的最高分。数据从文件score.txt读取。

解题思路：
需要从Score字段中找出最大值。A.max(x) 计算序列A中所有记录的x值的最大值，在序表或者排列中求最大值，和第9章在序列中求最大值的方法是类似的。

	A	B
1	=file("txt/scores.txt").import@t().select(Subject==1)	/加载科目1成绩数据
2	=A1.max(Score)	/99；返回最好成绩值

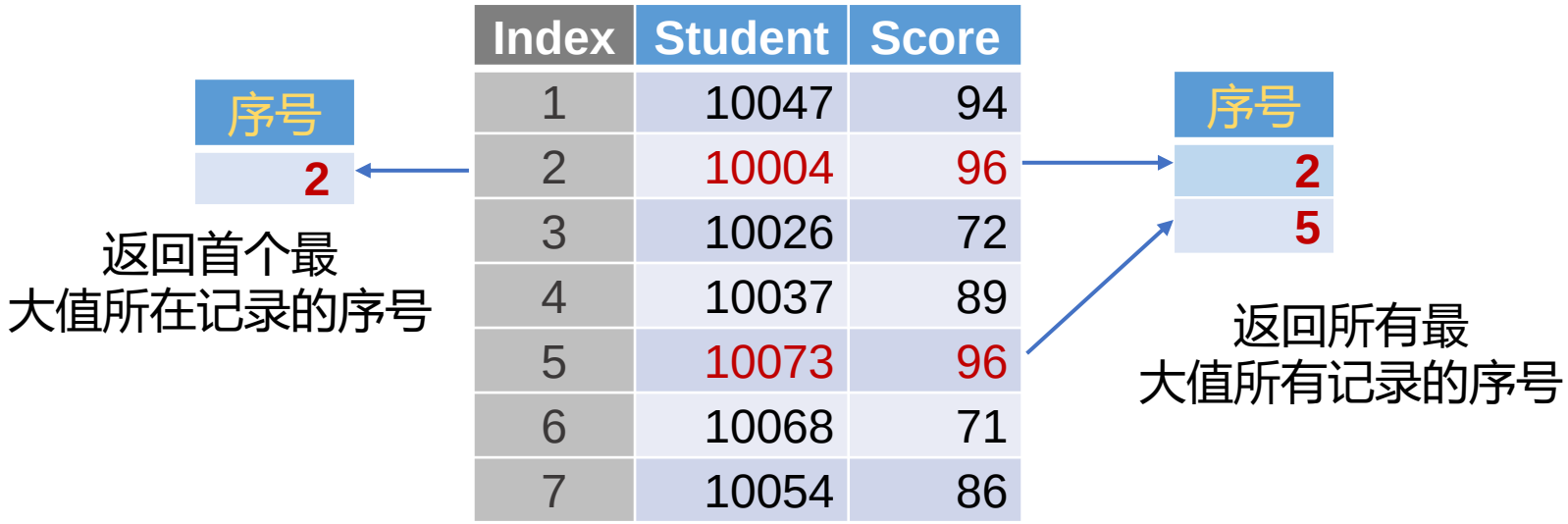
返回最大值：

A2	Value
	99

11.5.1 极值

B、找出最大值所在记录的序号。

查找最高分所在记录的序号：



11.5.1 极值

【例11.34】根据商品每日销售记录表，计算销量最大的记录增幅是多少。数据从文件goods.txt读取。

解题思路：

由于销售记录按日期排序，A.pmax(x)可获取销量最大的记录序号，通过序号得到对应上一次销售的记录，再计算增幅。

A.pmax(x) 计算x最大值所在记录的序号；缺省返回首个最大值所在记录的序号， 选项@a返回所有最大值所在记录的序号。

	A	B	C
1	=file("txt/goods.txt").import@t()	=create(SID, SALE, RATE)	/加载5月销售数据
2	=A1.pmax@a(SALE)		/获取所有最大销量所在记录的序号
3	=A1.calc(A2,(SALE/SALE[-1]-1))		/计算增幅
4	for A2	>B1.insert(0, A1(A4).SID, A1(A4).SALE, round(100*A3(A2.pos(A4)), 2)/"%")	/有关数据存为序表，round()将增幅保住两位小数

A1

Index	SID	PDATE	SALE
1	101	2019-03-01	260
2	102	2019-03-02	184
3	103	2019-03-03	328
4	104	2019-03-04	143
5	105	2019-03-05	210
…	…	…	…

A2

Index	Member
1	3
2	6

B1

Index	SID	SALE	RATE
1	103	328	78.26%
2	106	328	56.19%

11.5.1 极值

C、找出最大值所在的记录。

查找最高分所在的记录：

返回首个最高分所在记录

Index	Student	Score
1	10004	96

Index	Student	Score
1	10047	94
2	10004	96
3	10026	72
4	10037	89
5	10073	96
6	10068	71
7	10054	86

返回所有最高分所在记录

Index	Student	Score
1	10004	96
2	10073	96

11.5.1 极值

【例11.35】查找科目1成绩最好的学号。数据从文件scores.txt读取。

解题思路：
可用A.maxp() 获取成绩最好的记录。A.maxp(x) 查找x最大值所在的记录，缺省返回首个x最大值所在的记录， 选项@a返回所有x最大值所在的记录。

	A	B
1	=file("txt/scores.txt").import@t().select(Subject:1)	/加载科目1成绩数据
2	=A1.maxp@a(Score)	/返回所有成绩最好的记录
3	=A2.(Student)	/返回成绩最好的学号

A1

Index	Student	Subject	Score
1	10001	1	71
2	10002	1	87
3	10003	1	88
4	10004	1	53
5	10005	1	91
…	…	…	…

成绩最好的记录

A2

Index	Student	Subject	Score
1	10054	1	99
2	10066	1	99
3	10068	1	99
4	10087	1	99

A3

Index	Member
1	10054
2	10066
3	10068
4	10087

11.5.1 极值

查询集合中最小值与最大值类似，如：

A、 A.min(x)找出x的最小值。
B、 A.pmin(x)找出x最小值所在记录的序号。
C、 A.minp(x)找出x最小值所在的记录。

下面在表中查找最差成绩：

	A	B
1	=file("zh/txt/scores.txt").import@t().select(Subject==1)	/加载科目1成绩数据
2	=A1.min(Score)	/返回最差成绩值
3	=A1.pmin(Score)	/返回成绩最差的首条记录序号
4	=A1.minp(Score)	/返回成绩最差的首条记录

11.5.2 TOP

TOP-N问题。有时我们并不关注所有的成员的排序情况，尤其当数据量比较大时，往往只关注最大(小)N个成员情况。

如求集合中最大N个成员，也会遇到下面几种情况：
A、找出最大N个成员的值。
B、找出最大N个成员对应的记录。
C、找出最大N个成员的序号。

同理，求最小N个成员也会有类似上面的几种情况。

Index	Student	Score
1	10047	94
2	10004	96
3	10026	72
4	10037	89
5	10073	94
6	10068	71
7	10054	86

最好成绩前3名在上面3种操作下的结果

A返回值：

值
96
94
94

B返回记录构成的排列：

Index	Student	Score
1	10004	96
2	10047	94
3	10073	94

C返回序号：

序号
2
1
5

11.5.2 TOP

A、找出集合中最大(小)N个成员的**值**。

下面按Score字段排序后，获取前5个成绩值：

成绩最差前5名

Index	Score↑
1	71
2	72
3	86
4	86
5	89

原数据表

Index	Student	Score
1	10047	94
2	10004	96
3	10026	72
4	10037	89
5	10073	94
6	10068	71
7	10054	86
8	10034	86

成绩最好前5名

Index	Score↓
1	96
2	94
3	94
4	89
5	86

获取**最大**前N个值时，按**降序**排序；获取**最小**前N个值时，按**升序**排序。

11.5.2 TOP

A、找出集合中最大(小)N个成员的**值**。

【例11.36】查找科目1的前5名成绩。数据从文件scores.txt读取。

解题思路：

查找前5名成绩可用A.top(n, x)实现。

A.top(n, x), n=0，返回null; n>0，计算所有记录的x值中最小的n个；n<0，计算所有记录的x值中最大的n个。

n不可省略， x省略解释为~。

	A	B
1	=file("txt/scores.txt").import@t().select(Subject==1)	/加载科目1成绩数据
2	=A1.top(-5, Score)	/返回成绩最好前5个值

最好成绩数据：

A2	Index	Member
	1	99
	2	99
	3	99
	4	99
	5	98

注意：当有多个相同的最大值时，最大值不一定都返回了。上面的例子中若99分超过5个，也只返回前5个。

11.5.2 TOP

B、找出集合中最大(小) N个成员对应的**记录**

下面按Score字段排序后，获取前5条记录：

成绩最差前5名

Index	Student	Score↑
1	10068	71
2	10026	72
3	10034	86
4	10054	86
5	10037	89

原数据表

Index	Student	Score
1	10047	94
2	10004	96
3	10026	72
4	10037	89
5	10073	94
6	10068	71
7	10054	86
8	10034	86

成绩最好前5名

Index	Student	Score↓
1	10004	96
2	10047	94
3	10073	94
4	10037	89
5	10054	86

获取**最大**前N条记录时，按**降序**排序；获取**最小**前N条记录时，按**升序**排序。

11.5.2 TOP

【例11.37】科目1中分别查找成绩最差、最好的前5名。表数据从文件scores.txt读取。

解题思路：
获取前N个极值成员对应的记录可以用**A.top(n; x)**。
A.top(n; x) 和A.top(n,x)类似，但返回的不是x的值，而是对应的**记录**。

	A	B
1	=file("txt/scores.txt").import@t().select(Subject==1)	/加载科目1成绩数据
2	=A1.top(5; Score)	/返回成绩最差的前5名
3	=A1.top(-5; Score)	/返回成绩最好的前5名

成绩**最差**数据：

A2			
Index	Student	Subject	Score↑
1	10047	1	50
2	10004	1	53
3	10026	1	57
4	10037	1	60
5	10073	1	60

成绩**最好**数据：

A3			
Index	Student	Subject	Score↓
1	10066	1	99
2	10087	1	99
3	10068	1	99
4	10054	1	99
5	10034	1	98

11.5.2 TOP

C、找出集合中最大(小)N个成员的序号，以便用这些序号去做其它的计算。

下面按Score字段排序后，获取前5条记录的序号：

成绩最差前5名序号

Student	Score↑	ROW
10068	71	6
10026	72	3
10034	86	8
10054	86	7
10037	89	4

原数据表

Index	Student	Score
1	10047	94
2	10004	96
3	10026	72
4	10037	89
5	10073	94
6	10068	71
7	10054	86
8	10034	86

成绩最好前5名序号

ROW	Student	Score↓
2	10004	96
1	10047	94
5	10073	94
4	10037	89
8	10054	86

获取前N条**最大值**记录序号时，按**降序**排序；获取前N条**最小值**记录序号时，按**升序**排序。

11.5.2 TOP

【例11.38】根据商品每日销售记录表，计算销量最大的前3笔比上一日的增量。表数据从文件goods.txt读取。

解题思路：
获取销量最大的前3笔记录序号，再通过序号计算比上一日的增量。获取记录序号可使用ptop()。
A.ptop(n, x) 与A.top(n,x)类似，但不是返回x值，而是x值对应的记录序号。

	A	B
1	=file("txt/goods.txt").import@t().sort(PDATE)	/导入数据
2	=A1.ptop(-3, SALE)	/获取销量最大的前3笔记录序号
3	=A1.calc(A2,(SALE-SALE[-1]))	/计算增量

销量最大记录序号

A2	Index	Member
	1	6
	2	3
	3	1

销售增量

A3	Index	Member
	1	118
	2	144
	3	260

11.5.2 TOP

重复极值记录。记录中的最大值(或最小值)可能不止一个，需要列出所有的最大值(或最小值)。下面根据Score成绩，获取极值记录：

最小值

Index	Student	Score↑
1	10068	71
2	10034	71

原数据表

Index	Student	Score
1	10047	94
2	10004	96
3	10026	72
4	10037	89
5	10073	96
6	10068	71
7	10054	86
8	10034	71

最大值

Index	Student	Score↓
1	10004	96
2	10073	96

如某科考试成绩最好(差)的常出现多个现象。

11.5.2 TOP

【例11.39】科目1中分别列出成绩最差、最好的记录。表数据从文件scores.txt读取。

解题思路：
A.top(n; x) n为±1时返回全部最小或最大值对应的记录。

	A	B
1	=file("txt/scores.txt").import@t().select(Subject==1)	/加载科目1成绩数据
2	=A1.top(1; Score)	/返回成绩最差记录
3	=A1.top(-1; Score)	/返回成绩最好记录

A2 成绩最差数据：

Index	Student	Subject	Score↑
1	10047	1	50

A3 成绩最好数据：

Index	Student	Subject	Score↓
1	10066	1	99
2	10087	1	99
3	10068	1	99
4	10054	1	99

11.6 习题

11.6 习题

【习题11.1】查询人力资源部门工龄大于10年的员工数。数据从文件emp.txt读取。

解题思路：
工龄可根据入职时间获取，A.count(x)可根据条件过滤数据并计数处理。

	A	B
1	=file("txt/emp.txt").import@t()	/加载文本数据，返回序表
2	=A1.count(DEPT=="HR" && age(HIREDATE)>10)	/查询人力资源部门工龄大于10年的员工

输出员工数：

A2	Value
	14

11.6 习题

【习题11.2】查找不到20岁就入职的员工。数据从文件emp.txt读取。

解题思路：
入职时间与出生日期相差不到20年的即为不到20岁的员工。A.select(x)支持表达式查询。

	A	B
1	=file("txt/emp.txt").import@t()	/加载文本数据，返回序表
2	=A1.select(elapse@y(BIRTHDAY, 20)>=HIREDATE)	/过滤

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000
2	4	Emily	Smith	F	Texas	1985-03-07	2006-08-15	HR	7000
3	6	Matthew	Johnson	M	California	1984-07-07	2005-07-07	Sales	11000
4	7	Alexis	Smith	F	Illinois	1972-08-16	2002-08-16	Sales	9000
5	9	Victoria	Davis	F	Texas	1983-12-07	2009-12-07	HR	3000
…	…	…	…	…	…	…	…	…	…

出生后n年的日期，用**elapse@y(F, n)**计算。

11.6 习题

【习题11.3】根据商品每日销售记录表，找出累计销量达到1000时的日销售记录。表数据从文件goods.txt读取。

解题思路：
将表中的销售数据累计，若累计销量超过1000则中止，然后根据返回的序号获取构成销量1000的记录。A.pselect(x) 实现累计到达条件后就中止

	A	B
1	=file("txt/goods.txt").import@t()	0
2	=A1.pselect((B1=B1+SALE, B1>1000))	/返回值为6
3	=A1.to(A2)	/获取前6条记录

A2

Index	SID	PDATE	SALE
1	101	2019-03-01	260
2	102	2019-03-02	184
3	103	2019-03-03	328
4	104	2019-03-04	143
5	105	2019-03-05	210
6	106	2019-03-06	328
7	107	2019-03-07	254
8	108	2019-03-08	239

A3

Index	SID	PDATE	SALE
1	101	2019-03-01	260
2	102	2019-03-02	184
3	103	2019-03-03	328
4	104	2019-03-04	143
5	105	2019-03-05	210

A2：累计销量若达到1000时则返回当前记录序号。

11.6 习题

【习题11.4】按款号给衣服编号，期望如下图No字段所示。数据从clothing.txt读取。

ModelNumber	ColorCode	SizeCode	Num	No
LN02960	1020	004	1	1
LN02960	1020	005	2	1
LN02960	1020	006	2	1
LN02960	1020	007	2	1
LN02960	1020	008	3	1
LN02960	1020	010	2	1
LN02960	1020	012	1	1
LN02960	1020	014	1	1
LN03194	1051	003	1	2
LN03194	1051	004	1	2
LN03194	1051	005	2	2
LN03194	1051	006	2	2
LN03194	1051	007	2	2
LN03194	1051	008	2	2
LN03194	1051	010	2	2
LN03194	1051	009	1	2
LN03184	1051	003	1	3
LN03184	1051	004	1	3
LN03184	1051	005	2	3

解题思路：
衣服的编号根据ModelNumber顺序编号，由于衣服款号重复，我们需要先按款号排序，然后将它用**A.ranks@i(x)**去重排名即可得到每款衣服的编号。

	A	B
1	=file("clothing.txt").import@tq().sort(ModelNumber)	/加载文本数据，并按款号排序返回
2	=A1.derive(A1.ranks@i(ModelNumber)(#):No)	/去重排名后将排名写入新建的新表

11.6 习题

【习题11.5】成绩表中的各科成绩按高低排序。数据从scores.txt读取。

解题思路：
记录按科目升序、成绩由高到低排序，可用A.sort(x_i, ...)函数实现。

	A	B
1	=file("txt/scores.txt").import@t()	/导入学生成绩数据
2	=A1.sort(Subject,-Score)	/按学科升序、成绩降序排序

A1	Index	Student	Subject	Score
	1	10001	1	71
	2	10001	2	81
	3	10001	3	81
	4	10001	4	83
	5	10001	5	77
	6	10002	1	87
	7	10002	2	67
	8	10002	3	79
	9	10002	4	96
	…	…	…	…

A2排序

A2	Index	Student	Subject	Score
	1	10005	1	91
	2	10003	1	88
	3	10002	1	87
	4	10001	1	71
	5	10004	1	53
	6	10003	2	97
	7	10001	2	81
	8	10004	2	79
	9	10002	2	67
	…	…	…	…

11.6 习题

【习题11.6】产品表按NO相同，PID每5个依次递增，NO不同时PID也递增。期望如下图PID字段所示。数据从product.txt读取。

NAME	NO	PID	cnt
P3C0001	90560	P3CA001	0
P3C0002	90560	P3CA001	1
P3C0003	90560	P3CA001	2
P3C0004	90560	P3CA001	3
P3C0005	90560	P3CA001	4
P3C0006	90560	P3CA002	0
P3C0007	90560	P3CA002	1
P3C0008	90560	P3CA002	2
P3C0009	90560	P3CA002	3
P3C0010	90560	P3CA002	4
P3C0011	90560	P3CA003	0
P3C0012	90560	P3CA003	1
P3C0013	90561	P3CA004	0
P3C0014	90561	P3CA004	1
P3C0015	90561	P3CA004	2
P3C0016	90561	P3CA004	3
P3C0017	90561	P3CA004	4
P3C0018	90561	P3CA005	0
P3C0019	90561	P3CA005	1

解题思路：
期望生成PID字段，将同NO的记录分组，每5条生成1组，最后不足5条记录单分一组。为此，在处理时依次计算cnt值，当cnt为0时进行累计计数，并用计数与“P3CA”拼成字符串存为新的PID。
可以用**A.derive()**追加PID字段建立新表，**A.run()**考察NO变化递增计数存入PID。

	A	B
1	=file("product.txt").import@tq().sort(NO)	/加载文本数据，并排序返回
2	=A1.derive(PID)	/追加字段PID到新表
3	>cnt=0, idx=0	
4	>A2.run(a= (if(NO==NO[-1], (cnt=cnt+1)%5, cnt=0)), PID=format("P3CA%03d", (if (a==0, idx=idx+1, idx))))	/计算求余计数

提示：调试过程中可以用derive()增加一个辅助的TMP字段来存储A3中的临时变量a，有助于观察分析问题。

11.6 习题

【习题11.6】产品表按NO相同，PID每5个依次递增，NO不同时PID也递增。

下面将A4分解成两步，并增加了TMP字段来说明如何获取PID值的。

NAME	NO	TMP	PID
P3C0001	90560	0	P3CA001
P3C0002	90560	1	P3CA001
P3C0003	90560	2	P3CA001
P3C0004	90560	3	P3CA001
P3C0005	90560	4	P3CA001
P3C0006	90560	0	P3CA002
P3C0007	90560	1	P3CA002
P3C0008	90560	2	P3CA002
P3C0009	90560	3	P3CA002
P3C0010	90560	4	P3CA002
P3C0011	90560	0	P3CA003
P3C0012	90560	1	P3CA003
P3C0013	90561	0	P3CA004
P3C0014	90561	1	P3CA004
P3C0015	90561	2	P3CA004
P3C0016	90561	3	P3CA004
P3C0017	90561	4	P3CA004
P3C0018	90561	0	P3CA005
P3C0019	90561	1	P3CA005

2	=A1.derive(TMP, PID)	/追加字段PID，TMP
3	>cnt=0, idx=0	/cnt计数, idx计数
4	>A2.run(TMP= (if(NO==NO[-1], (cnt=cnt+1)%5, cnt=0)))	/计算求余计数
5	>A2.run(PID=format("P3CA%03d", (if (TMP==0, idx=idx+1, idx))))	

把A4分解两步：

1、**计算5的余数**。A4中当相邻的NO相同时，计数变量cnt累计加1，cnt按5的余数写入TMP；遇到NO不同时，则cnt重置为0写入TMP。计算余数产生的结果如TMP列所示。

2、**组成字符串**。A5中根据TMP值生成PID数据。当遇到TMP值为0时，则计数变量idx累计加1，否则idx值不变， idx与“P3CA”组成新字符串写入PID。生成结果如PID列所示。

11.6 习题

【习题11.7】某年级各班卫生成绩情况如下，求每行卫生成绩由高到低排序并增加班级名显示。数据从hygiene.txt读取。

Date	Class1	Class2	Class3	Class4	Class5
2020-01-02	89	82	95	93	96
2020-01-03	93	90	98	93	97
2020-01-04	95	94	95	99	92
2020-01-05	97	89	93	93	95
2020-01-06	94	96	95	94	91

Date	No1	No2	No3	No4	No5
2020-01-02	Class5(96)	Class3(95)	Class4(93)	Class1(89)	Class2(82)
2020-01-03	Class3(98)	Class5(97)	Class1(93)	Class4(93)	Class2(90)
2020-01-04	Class4(99)	Class1(95)	Class3(95)	Class2(94)	Class5(92)
2020-01-05	Class1(97)	Class5(95)	Class3(93)	Class4(93)	Class2(89)
2020-01-06	Class2(96)	Class3(95)	Class1(94)	Class4(94)	Class5(91)

解题思路：

1、从序表中获取班级字段。

2、将成绩与对应的班级字段结合成序列A。

3、将日期与序列A合并后写入新序表。

11.6 习题

【习题11.7】某年级各班卫生成绩情况如下，求每行卫生成绩由高到低排序并增加班级名显示。数据从hygiene.txt读取。

	A	B
1	=file("hygiene.txt").import@t().sort(Date)	/加载文本数据
2	=A1.fname().to(2,)	/获取表中班级字段信息
3	=A1.((a=~.array().to(2,), b=a.psort@z(), b.(A2(~)/ "(" / a(~) / ")")))	/成绩与班级组合
4	=A3.conj(A1(#).Date|~)	/日期与A3中的数据合并
5	=create(Date, No1, No2, No3, No4, No5).record(A4)	/合并数据写入序表

A2

Index	Member
1	Class1
2	Class2
3	Class3
4	Class4
5	Class5

A3

Index	Member
1	[Class5(96), Class3(95), Class4(93), …]
2	[Class3(98), Class5(98), Class1(93), …]
3	[Class4(99), Class1(95), Class3(95), …]
4	[Class1(97), Class5(95), Class3(93), …]
5	[Class2(96), Class3(95), Class1(94), …]

A4

Index	Member
1	2020-01-02
2	Class5(96)
3	Class3(95)
4	Class4(93)
…	…

A5

Index	Date	No1	…
1	2020-01-02	Class5(96)	…
2	2020-01-03	Class3(98)	…
3	2020-01-04	Class4(99)	…
4	2020-01-05	Class1(97)	…
5	2020-01-06	Class2(96)	…

说明： A3中a为记录的成绩序列，b为成绩倒排列后的序号，然后通过序号将成绩与班级信息组合。

11.6 习题

【习题11.8】查询销售ID为10的商家销量最大的5笔订单，与前一次订单比较分别增加了多少?

解题思路：
A.ptop()找到销量最大的5笔订单的记录序号，由返回的序号可获取对应的订单及对应的前一次订单的记录，从而对两次记录作相应的计算。其中A.(x)计算前一次订单记录序号，A.derive()存储订单的增量

	A	B	C
1	=file("txt/sales.txt").import@t().select(SELLERID==10)	=A1.sort(ORDERDATE)	/加载文本数据，返回排列
2	=B1.ptop(-5; AMOUNT)	=B1(A2)	/查询销量最大前5条记录
3	=A2.(B1.m(~-1).AMOUNT)		/由序号找到前一次的订单记录
4	=B2.derive(A3(#): PREMOUNT, AMOUNT-PREMOUNT: DIFF)		/将相关记录写入新序表

A4

Index	ORDERID	CLIENT	SELLERID	AMOUNT	ORDERDATE	PREMOUNT	DIFF
1	29	QHHW	10	29500.0	2012-12-05	21800.0	7700.0
2	361	HANAR	10	28500.0	2013-11-04	4900.0	23600.0
3	259	EGU	10	25600.0	2013-07-18	23700.0	1900.0
4	433	DNEDL	10	24200.0	2014-01-07	5390.0	18810.0
5	182	VILJX	10	23700.0	2013-05-05	7448.0	16252.0

通过记录序号，可获取相邻数据并了解它们之间的变化。

11.6 习题

【习题11.9】查询600005股票的最大连涨天数的记录。数据从文件stockrecords.txt读取。

解题思路：
根据每日股价差额用**A.pmax()**找到最大连涨天数的位置，计算出其最大天数，**A.to()**获取构成此数据的记录。

	A	B
1	=file("txt/stockrecords.txt").import@tq().select(STOCKID=="600005").sort(DATE)	/加载文本数据，按交易日期排序返回排列
2	>z=0, max=0	/max记录最大连涨天数，z记录当前连涨天数
3	=A1.pmax(if(CLOSING-CLOSING[-1]>0,(z=z+1,if (z>max, max=z)), (if (z>max, max=z), z=0)))	/计算出最大连涨天数，返回最大连涨天数序号
4	=A1.to(A3-max, A3)	/获取最大连涨天数相关的记录

A5

Index	STOCKID	DATE	CLOSING
1	600005	2015-11-20	19.98
2	600005	2015-11-23	20.69
3	600005	2015-11-24	22.5
4	600005	2015-11-25	23.4
…	…	…	…

说明：A3中第2个if计算当前连涨天数，第3个if计算最大连涨天数，若无第3个if计算，如果最大连涨出现在A1最后一个，就没可能把z设给max。

第12章 有序计算

12.1 有序集合

- 有序集合的概念
- 序号访问

12.2 迭代函数

12.3 定位计算

- 定位成员
- 定位最大值/最小值
- 获取满足条件的成员序号

12.4 排序与对齐

- 排序

12.5 习题

12.1 有序集合

12.1.1 有序集合的概念

有序集合是一组有顺序的数据。相当于某些程序语言中的数组或链表。有序集合的成员可以重复。

12.1.1 有序集合的概念

有序集合与**集合（无序）**的区别如下：

- **有序集合**的**成员可以重复，**提供了按序号访问成员的方式。（注意有序是指成员的位置有序，而不是排序。）

- **集合（无序）**的**成员不可重复**　只能遍历，无法通过序号访问。

12.1.2序号访问

序号访问是指通过序号（索引下标）来访问有序集合的成员。

序号访问的常见类型有哪些?

- 直接序号访问
- 区间序号访问
- 固定跨度访问
- 平均分段访问

12.1.2序号访问

直接序号访问是指直接通过单个或者多个序号来访问有序集合的成员。

► **单序号访问**

第3个成员 →

序号	成员
1	a
2	b
3	c
4	d
5	e
6	f

► **多序号访问**

第2、3、5个成员 →

序号	成员
1	a
2	b
3	c
4	d
5	e
6	f

12.1.2序号访问

【例12.1】查询2019年1月上证指数第一个交易日和最后一个交易日的收盘价。 1月收盘价的全部数据如下：

[2465.291,2464.3628,2514.8682,2533.0887,2526.4622,2544.3449,2535.0985,2553.8313,2535.7653,2570.3449,2570.4221,2559.637,2596.0056,2610.5094,2579.7041,2581.0041,2591.6935,2601.7234,2596.9767,2594.2536,2575.5756,2584.5724]

12.1.2序号访问

【例12.1】查询2019年1月上证指数第一个交易日和最后一个交易日的收盘价。

1	2	3	…	20	21	22
2465.291	2464.3628	2514.8682	…	2594.2536	2575.5756	2584.5724

解题思路

可以直接使用序号访问有序集合的成员。

	A	B	C	D
1	[2465.291, 2464.3628, …, 2575.5756,2584.5724]			/2019年1月收盘价的常数序列
2	=A1(1)	=A1.m(1)	/负数序号表示倒数	/访问第1个成员
3	=A1(22)	=A1.m(22)	=A1.m(-1)	/访问第22个（倒数第1个）成员
4	=A1([1,22])	=A1.m([1,22])	=A1.m([1,-1])	/多序号访问，第1个和第22个（倒数第1个）成员

执行结果：

2465.291

2584.5724

[2465.291,2584.5724]

A1中定义了一个**常数序列**，**序列**是由数据成员构成的有序集合。序列成员用逗号分隔，中括号括起来。由于篇幅关系，A1中的常数**序列**做了省略，完整数据在题目中已经给出。

12.1.2序号访问

直接序号访问的方法：

- **单序号访问**：对于有序集合A的第i个成员，可以使用：A(i)，A.m(i) 来访问。
- **多序号访问**：对于有序集合A的第$[i_1,i_2,...,i_n]$个成员，可以使用：A($[i_1,i_2,...,i_n]$), A.m($[i_1,i_2,...,i_n]$) 同时访问多个成员
- A.m()函数可以支持负数序号，表示倒数。比如A.m(-1)表示取倒数第一个成员，A.m([-1,-2])表示取倒数第一个和第二个成员。

其中$[i_1,i_2,...,i_n]$表示由成员序号组成的序列。

12.1.2序号访问

区间序号访问是指通过序号区间来访问有序集合的成员。

第2到5个成员

序号	成员
1	a
2	b
3	c
4	d
5	e
6	f

倒数第1到3个成员

序号	成员
1	a
2	b
3	c
4	d
5	e
6	f

【例12.2】查询2019年1月上证指数前3个交易日和最后3个交易日的收盘价。1月收盘价的全部数据如下:

[2465.291,2464.3628,2514.8682,2533.0887,2526.4622,2544.3449,2535.0985,2553.8313,2535.7653,2570.3449,2570.4221,2559.637,2596.0056,2610.5094,2579.7041,2581.0041,2591.6935,2601.7234,2596.9767,2594.2536,2575.5756,2584.5724]

12.1.2序号访问

【例12.2】查询2019年1月上证指数前3个交易日和最后3个交易日的收盘价。

1	2	3	…	20	21	22
2465.291	2464.3628	2514.8682	…	2594.2536	2575.5756	2584.5724

解题思路

可以使用区间序号访问有序集合的成员。

	A	B	C	D
1	[2465.291, 2464.3628, …, 2575.5756,2584.5724]			/2019年1月收盘价的常数序列
2	=A1.to(3)	=A1.m(1:3)	=A1.to(1,3)	/访问前3个成员
3	=A1.to(-3)	=A1.m(-3:-1)		/访问最后3个成员
4	=A1.m(1:3,-3:-1)			/访问前3个和最后3个成员

执行结果：

[2465.291,2464.3628,2514.8682]

[2594.2536,2575.5756,2584.5724]

[2465.291,2464.3628,2514.8682,
2594.2536,2575.5756,2584.5724]

想要查询第3个到第5个交易日的收盘价：

	A	B	C
2	=A1.to(3,5)	=A1.m(3:5)	/访问第3到5区间的成员
3	=A1.to(5,3)		/访问第5到3区间的成员，成员顺序与A2结果相反

[2514.8682,2533.0887,2526.4622]

[2526.4622,2533.0887,2514.8682]

12.1.2序号访问

序号区间访问的方法：

- 前n个/最后n个：对于有序集合A的前n个成员，可以使用：A.to(n)来访问；最后n个成员，可以使用：A.to(-n) 来访问
- 单区间访问：对于有序集合A的第a个到第b个成员，
  当a<=b时可以使用：A.m(a:b) 和 A.to(a,b) 函数；
  当a>=b时也可以使用： A.m(a:b) 和 A.to(a,b) 函数，但返回成员的序号顺序将从a到b依次递减。
- 多区间和序号访问：可以使用 A.m() 函数访问多个区间和序号。例如：A.m(a:b,[c,d],e, ...) 。

12.1.2序号访问

固定跨度访问是指按指定的起始序号和固定的跨度访问成员。

例如要取1到10之间的偶数，从第二个成员开始取，跨度为2取数，得到 [2,4,6,8,10]：

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

【例12.3】从2到100以内的偶数列，取其中5的倍数。

12.1.2序号访问

【例12.3】从2到100以内的偶数列，取其中5的倍数。

1	…	5	…	10	…	15	…	50
2	…	10	…	20	…	30	…	100

解题思路

取倍数可以按固定跨度访问成员。

	A	B
1	=to(100).step(2,2)	/从第2个成员起，每2个成员取数，即100以内的偶数列
2	=A1.step(5,5)	/从第5个成员起，每5个成员取数

执行结果：

[2,4,6,8,10,12,14,16,18,20,…,100]

[10,20,30,40,50,60,70,80,90,100]

注意

- to(n)函数用来创建1到n的序列。to(100).step(2,2)可以简化为100.step(2,2) 。
- 有序集合A从序号k开始，按固定跨度m访问成员，可以使用：A.step(m,k)函数。

12.1.2序号访问

平均分段访问是指将有序集合平均分段，访问其中的一段。

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

第1段　　第2段

不能平分时，后面的分段数会比前面的分段少1个成员：

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

第1段　　第2段　　第3段

【例12.4】从2到100以内的偶数列，取前五分之一和后五分之一的成员。

12.1.2序号访问

【例12.4】从2到100以内的偶数列，取前五分之一和后五分之一的成员。

1	…	10	11	…	40	41	…	50
2	…	20	22	…	80	82	…	100

解题思路

可以采用平均分段的方式，来获取有序集合中的某段区间内的成员。

	A	B
1	=100.step(2,2)	/2到100的偶数序列
2	=A1.to@z(1,5)	/将有序集合平均分为5段，取第1段
3	=A1.to@z(5,5)	/将有序集合平均分为5段，取第5段。

执行结果：

[2,4,6,8,10,12,14,16,18,20]

[82,84,86,88,90,92,94,96,98,100]

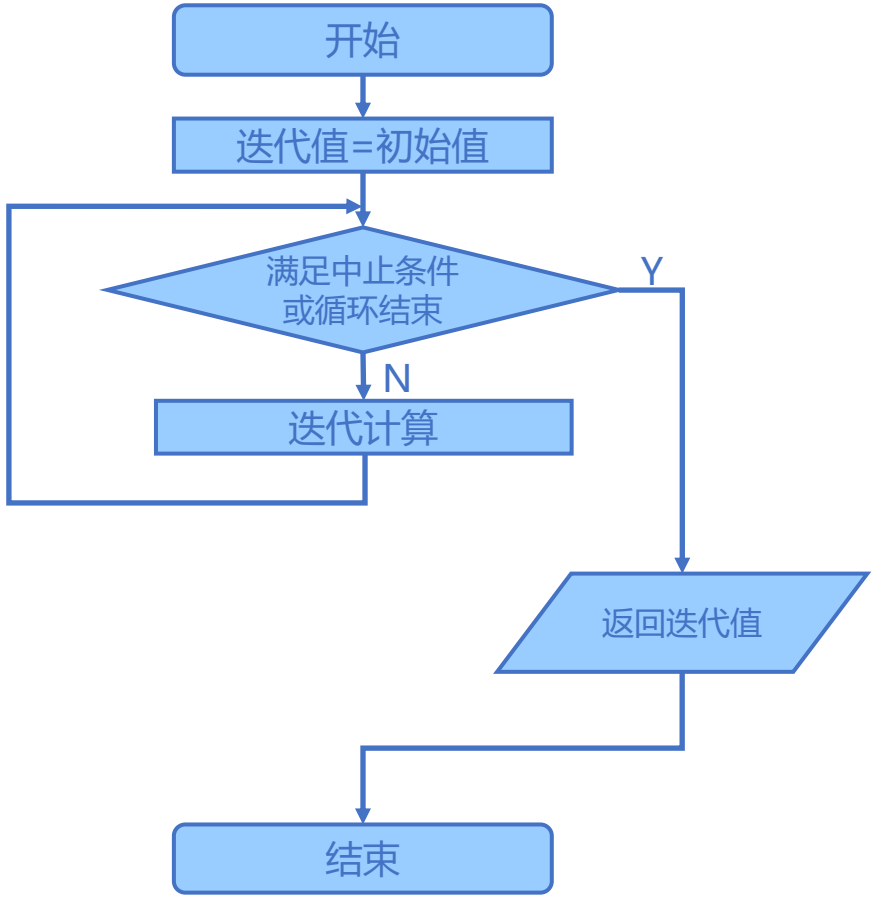
注意

- 将有序集合A平均分成n段，取其中的第i段，可以使用A.to@z(i,n)函数。

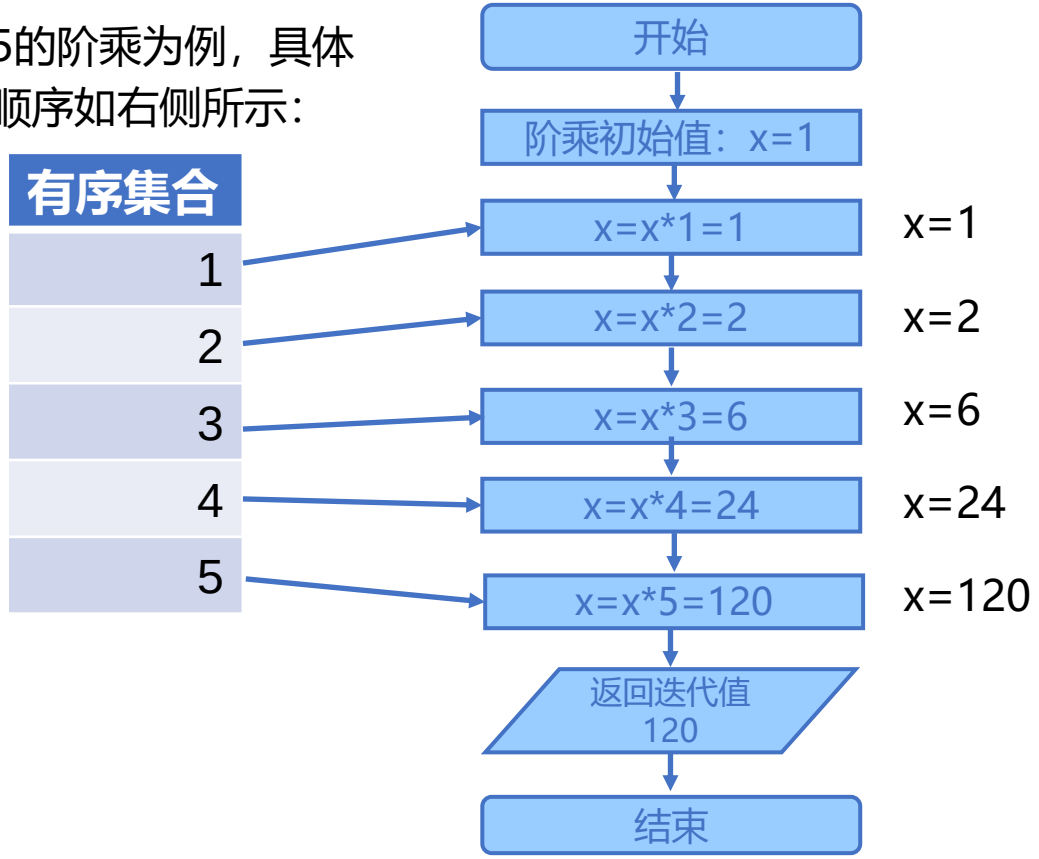
12.2 迭代函数

12.2 迭代函数

迭代是指重复执行运算，直到满足某条件为止。用于实现迭代计算的函数我们称之为**迭代函数**。



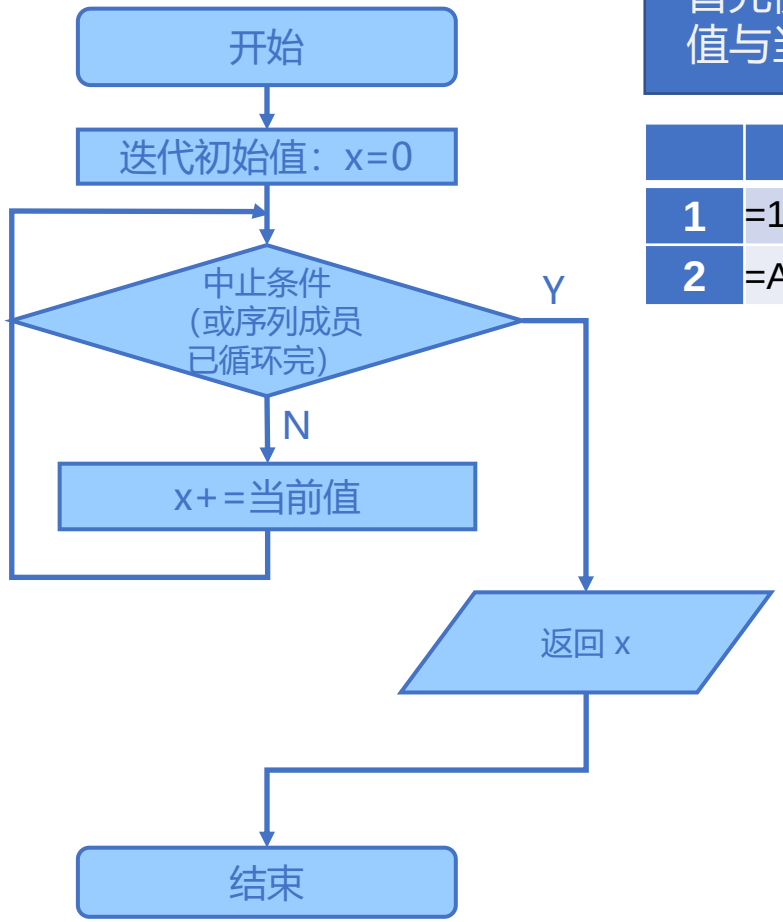
以求5的阶乘为例，具体执行顺序如右侧所示：



【例12.5】等差数列求和：1+3+5+7+9+13+...+99。

12.2 迭代函数

【例12.5】等差数列求和：1+3+5+7+9+13+...+99。



解题思路

首先使用跨度取数创建100以内的奇数列。再使用迭代函数，依次计算迭代值与当前数字的和。

	A	B
1	=100.step(2, 1)	/按跨度取数，获得100以内奇数列
2	=A1.iterate(~~+~, 0)	/迭代计算，累加当前数值，初始值为0

A2
2500

函数 A.iterate(x,a,c) 可以用于迭代计算。

A.iterate(x,a,c)的参数和返回值

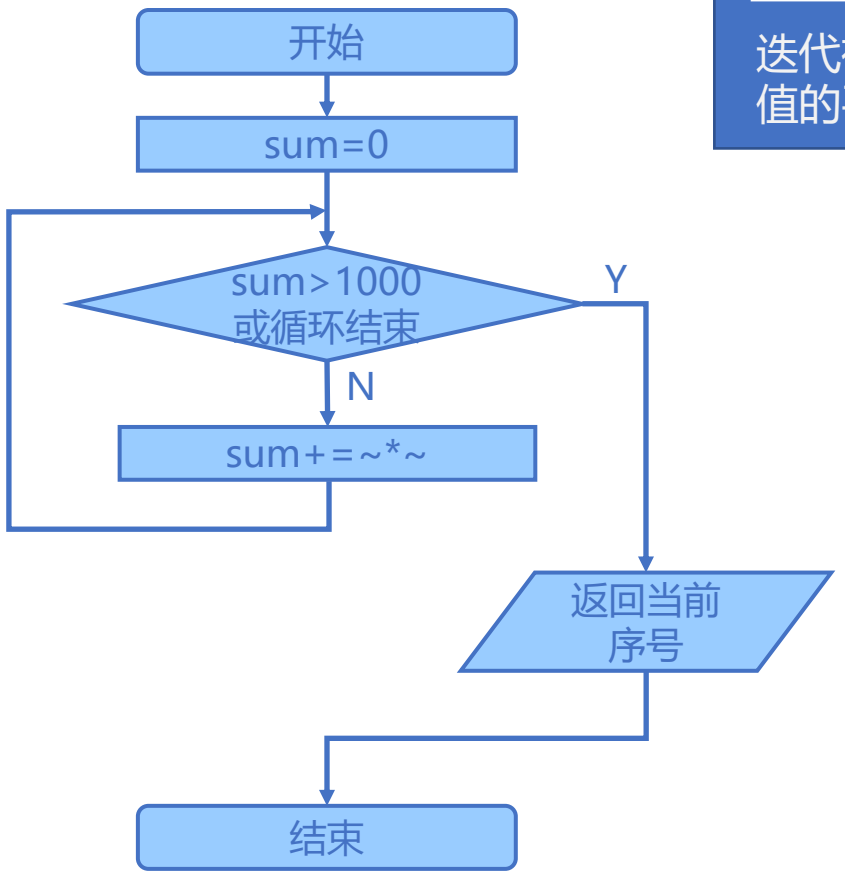
- x：含有~~的迭代表达式。
- a：迭代初始值。
- c：迭代中止的条件表达式，省略时迭代到循环结束。
- ~~：返回值。

函数执行时，迭代开始前初始值为a，并将其赋值给迭代值~~，每次先判断c，如果为真且未循环完A则计算x，并用~~来保存结果，迭代完成或c为真则返回~~。

【例12.6】求n，使得1到n的平方和刚好超过1000。

12.2 迭代函数

【例12.6】求n，使得1到n的平方和刚好超过1000。



解题思路

迭代初始值为0，从1至1000（n一定小于1000）循环迭代，依次累加当前数值的平方。即$1^2+2^2+...+n^2$，当结果大于1000时终止迭代。

	A	B
1	>sum=0	/定义变量sum，代表平方和。
2	=1000.iterate((sum+=power(~,2), ~~=#),0,sum>1000)	/循环累加每个成员的平方和，当累计超过1000时终止，返回当前序号。

A2
14

注意

- 这里的返回值是序号，不是平方和。所以另外定义变量sum用来计算平方和，把当前序号#赋值给~~。

12.3 定位计算

12.3.1 定位计算

定位计算是指在有序集合中，根据指定条件获取成员的序号。

定位计算的常见类型有哪些?

- 定位成员
- 定位最大值/最小值
- 获取满足条件的成员序号

12.3.1 定位计算

前面介绍了如何通过**序号**访问有序集合的**成员**。**定位成员**正好相反，要在有序集合中通过比较**成员**来查找**序号**。

定位名字是"Emily"的员工。

Emily的序号为4

ID	NAME	BIRTHDAY	ENTRYDATE	DEPT	SALARY
1	Rebecca	1974/11/20	2005/03/11	R&D	7000
2	Ashley	1980/07/19	2008/03/16	Finance	11000
3	Rachel	1970/12/17	2010/12/01	Sales	9000
4	Emily	1985/03/07	2006/08/15	HR	7000
5	Ashley	1975/05/13	2004/07/30	R&D	16000

依次比较每个员工名字与"Emily"是否相同，相同时返回序号。

12.3.1 定位计算

【例12.7】查找成绩100分的序号。全部成绩如下：

[94,97,100,96,96,92,99,87,82,92,87,89,97,90,99,95,93,100,96,98,88,90,90,88,89,97,99,93,100,82]

12.3.1 定位计算

【例12.7】查找成绩100分的序号。全部成绩如下：

1	2	3	…	28	29	30
94	97	100	…	93	100	82

解题思路

在有序集合中定位成员，依次比较成员与100分是否相等，返回相等的序号。

	A	B
1	[94,97,100,96,96,92,99,87,82,92,87,89,97,90,99,95,93,100,96,98,88,90,90,88,89,97,99,93,100,82]	/分数的常数序列
2	=A1.pos(100)	/使用pos()函数，定位100分所在序号。

执行结果：

3

可以从后向前定位成员：

	A	B
2	=A1.pos@z(100)	/使用pos()函数的@z选项，从后向前定位100分所在序号。

29

100分可能不止一个：

	A	B
2	=A1.pos@a(100)	/使用pos()@a函数定位所有100分所在序号。

[3,18,29]

12.3.1 定位计算

定位成员的方法：

- 对于有序集合A，定位成员x ，可以使用A.pos(x)函数在有序集合中通过成员来获取序号。
- 有序集合的成员可以重复，定位成员时可能返回多个序号。A.pos@a(x)可以返回所有x的序号，默认只返回第一个。
- 定位成员默认是正向定位，需要逆向定位时可以使用：A.pos@z(x) 。

定位最大值/最小值，是指查找成员最大值/最小值所在的序号。

定位工资最高和最低的员工。

ID	NAME	BIRTHDAY	ENTRYDATE	DEPT	SALARY
1	Rebecca	1974/11/20	2005/03/11	R&D	7000
2	Ashley	1980/07/19	2008/03/16	Finance	11000
3	Rachel	1970/12/17	2010/12/01	Sales	9000
4	Emily	1985/03/07	2006/08/15	HR	6500
5	Ashley	1975/05/13	2004/07/30	R&D	16000

最低工资序号为4

最高工资序号为5

【例12.8】计算2019年上证指数收盘价最高点和最低点当天的涨跌幅。全部数据如下：

[2465.291,2464.3628,2514.8682,2533.0887,2526.4622,2544.3449,2535.0985,2553.8313,2535.7653,2570.3449,2570.4221,2559.637,2596.0056,2610.5094,2579.7041,2581.0041,2591.6935,2601.7234,2596.9767,2594.2536,2575.5756,2584.5724,2618.2323,2653.896,2671.8937,2721.0683,2719.6999,2682.385,2754.3566,2755.6459,2761.2189,2751.8012,2804.2262,2961.2834,2941.5163,2953.8242,2940.9538,2994.005,3027.5755,3054.2466,3102.0991,3106.4179,2969.8614,3026.9925,3060.3073,3026.9513,2990.685,3021.7512,3096.4176,3090.975,3090.6403,3101.455,3104.1487,3043.0313,2997.0954,3022.7196,2994.9434,3090.758,3170.361,3176.8227,3216.2952,3246.5714,3244.8103,3239.6639,3241.9299,3189.9619,3188.6256,3177.7866,3253.5978,3263.1179,3250.2012,3270.7973,3215.0426,3198.5938,3201.6137,3123.8261,3086.3989,3062.4985,3078.3389,2906.464,2926.3908,2893.7567,2850.9537,2939.2106,2903.7131,2883.6107,2938.6781,2955.711,2882.2962,2870.6046,2905.9688,2891.7046,2852.5153,2852.9948,2892.3782,2909.9106,2914.6964,2905.8054,2898.6961,2890.0809,2862.2803,2861.4181,2827.7978,2852.1302,2925.7162,2909.3796,2910.7406,2881.9743,2887.6221,2890.158,2917.8029,2987.1186,3001.9802,3008.1479,2982.0738,2976.2837,2996.7926,2978.8784,3044.9028,3043.9428,3015.2633,3005.2488,3011.0588,2933.3634,2928.2295,2915.3022,2917.7609,2930.5462,2942.1853,2937.6168,2931.6926,2901.1766,2924.2006,2886.974,2899.9446,2923.2775,2937.3601,2944.541,2941.007,2952.3383,2932.5058,2908.766,2867.8376,2821.4957,2777.5559,2768.6795,2794.5523,2774.7532,2814.9944,2797.2596,2808.9146,2815.7976,2823.8238,2883.096,2880.0012,2880.3307,2883.4358,2897.4253,2863.5673,2902.1932,2893.7564,2890.9192,2886.2365,2924.1063,2930.1538,2957.4116,2985.8648,2999.6013,3024.7388,3021.2024,3008.8118,3031.2351,3030.7544,2978.1178,2985.6586,2999.2789,3006.4467,2977.0767,2985.3406,2955.4325,2929.0875,2932.167,2905.1892,2913.5704,2924.8566,2947.7106,2973.6558,3007.8834,2991.0459,2978.7124,2977.3342,2938.1413,2939.6179,2954.3799,2941.6188,2940.9215,2954.9327,2980.0498,2954.176,2939.3209,2929.0561,2958.1992,2975.4919,2991.5622,2978.5955,2978.7144,2964.1849,2909.9746,2914.8232,2905.2409,2909.8697,2891.3431,2909.2002,2933.9908,2911.0534,2903.6379,2885.2884,2906.1688,2907.0602,2903.1947,2889.6934,2871.9813,2875.8072,2884.6988,2878.1157,2899.4685,2912.0136,2914.4775,2917.3157,2924.4173,2915.6983,2967.6764,2984.3925,3022.4219,3017.0444,3017.0658,3004.9376,2962.7513,2982.6806,2981.8805,3007.3546,3005.0355,3040.0239,3050.124]

12.3.1 定位计算

【例12.8】计算2019年上证指数收盘价最高点和最低点当天的涨跌幅。

1	2	3	…	242	243	244
2465.291	2464.3628	2514.8682	…	3005.0355	3040.0239	3050.124

解题思路

先定位最大值/最小值的序号，再根据序号取出当日和前一日的收盘价，计算当天的涨幅。

	A	B
1	[2465.291, 2464.3628, …, 3040.0239, 3050.124]	/2019年上证指数收盘价常数序列，全部数据见例题
2	=A1.pmax()	/使用pmax()函数，定位收盘价最高点所在序号。
3	=if(A2>1, A1(A2)/A1(A2-1), 1.0)	/利用最高点的序号，计算最高点与前日的涨跌幅。如果首日收盘价最高，则认为涨幅为1.0
4	=A1.pmin()	/使用pmin()函数，定位收盘价最低点所在序号。
5	=if(A4>1, A1(A4)/A1(A4-1), 1.0)	/利用最低点的序号，计算最低点与前日的涨跌幅。如果首日收盘价最低，则认为涨幅为1.0

执行结果：

72
1.0063
2
0.9996

12.3.1 定位计算

定位最大值/最小值的方法：

- 对于有序集合A，可以使用A.pmax()函数**定位最大值**，可以使用A.pmin()函数**定位最小值**。
- 最大值/最小值可能有多个，A.pmax@a()/ A.pmin@a()可以返回所有最大值/最小值的序号，默认只返回第一个。
- 定位最大值/最小值默认是正向，需要逆向定位时可以使用A.pmax@z()/A.pmin@z() 。

获取满足条件的成员序号，是指对有序集合的成员按指定条件进行计算，返回结果为真的成员序号。

定位工资超过1万的员工。

ID	NAME	BIRTHDAY	ENTRYDATE	DEPT	SALARY
1	Rebecca	1974/11/20	2005/03/11	R&D	7000
2	Ashley	1980/07/19	2008/03/16	Finance	11000
3	Rachel	1970/12/17	2010/12/01	Sales	9000
4	Emily	1985/03/07	2006/08/15	HR	6500
5	Ashley	1975/05/13	2004/07/30	R&D	16000

工资超过1万元的序号为[2,5]

12.3.1 定位计算

【例12.9】求2019年上证指数收盘价比开盘价上涨超过80点的交易日，当天的交易量涨幅。数据从文件SSEC.txt中读取：

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
…	…	…	…

12.3.1 定位计算

【例12.9】求2019年上证指数收盘价比开盘价上涨超过80点的交易日，当天的交易量涨幅。数据从文件SSEC.txt中读取：

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
…	…	…	…

解题思路

可以先获取满足条件（Close-Open>80）的成员序号，再根据序号取出当日和前一日的交易量，计算当天的涨幅。

	A	B
1	=file("SSEC.txt").import@t()	/导入上证指数数据
2	=A1.select(year(Date)==2019).sort(Date)	/使用select()函数选出2019年的记录。使用sort()函数按日期升序排列
3	=A2.pselect@a(Close-Open>80)	/使用pselect@a()函数，定位所有收盘价与开盘价超出80点的序号。
4	=A3.new(A2(~).Date:Date, if(~>1, A2(~).Amount/A2(~-1).Amount, 1.0):Increase)	/使用A.new()函数创建序表。循环定位的序号，计算这几个交易日与前日的交易量涨幅。对于首日认为涨幅为1.0

A4

Date	Increase
2019/02/25	1.758491
2019/03/29	1.334483
2019/04/16	1.011236

获取满足条件的成员序号的方法：

- 对于有序集合A，可以使用A.pselect(x)函数获取满足条件x的成员序号。
- 满足条件x的成员可能有多个，A.pselect@a(x)可以返回所有满足条件x的序号，默认只返回第一个。
- 获取满足条件的成员序号默认是正向定位，需要逆向定位时可以使用A.pselect@z(x) 。

12.4 排序与对齐

12.4.1 排序

将有序集合的成员，通过一定的方法按关键字顺序排列的过程叫做**排序**，其目的是将一组数据**“无序”**的序列调整为数据**“有序”**的序列。

员工表

ID	NAME	SALARY
1	Rebecca	7000
2	Ashley	11000
3	Rachel	9000
4	Emily	6500

按工资从低到高排序。

ID	NAME	SALARY
4	Emily	6500
1	Rebecca	7000
3	Rachel	9000
2	Ashley	11000

按工资从高到低排序。

ID	NAME	SALARY
2	Ashley	11000
3	Rachel	9000
1	Rebecca	7000
4	Emily	6500

12.4.1 排序

【例12.10】将上证指数按交易额升序排列。数据从文件SSEC.txt中读取：

Date	Amount
2019/12/31	2.27E11
2019/12/30	2.67E11
2019/12/27	2.58E11
2019/12/26	1.96E11
…	…

12.4.1 排序

【例12.10】将上证指数按交易额升序排列。数据从文件SSEC.txt中读取：

解题思路

对有序集合按照指定字段（交易额）进行升序排列。

	A	B
1	=file("SSEC.txt").import@t()	/导入数据文件
2	=A1.sort(Amount)	/使用sort()函数按交易额升序排列

Date	Amount
2019/12/31	2.27E11
2019/12/30	2.67E11
2019/12/27	2.58E11
2019/12/26	1.96E11
…	…

A2

Date	Amount
2019/01/02	9.76E10
2019/01/30	1.01E11
2019/01/23	1.05E11
…	…

注意

- sort(x) 函数可以按照指定条件x进行排序。默认升序排列，使用z选项可以逆序。

12.4.1 排序

有的时候，不想改变**有序集合**的原序，只想获取**排序**后的**序号**。

员工表已经按名字有序，不改变员工表，求员工按工资升序排列后的原序号。

NAME	SALARY
Ashley	11000
Emily	6500
Rachel	9000
Rebecca	7000

工资排名序号
2
4
3
1

排名第1的员工序号是2

排名第2的员工序号是4

排名第3的员工序号是3

排名第4的员工序号是1

12.4.1 排序

【例12.11】按员工工资升序顺序取出员工姓名。员工表从文件EmployeeSample.txt中读取：

ID	Name	BirthDay	EntryDate	Department	Salary
1	Rebecca	1974/11/20	2005/03/11	6	7000
2	Ashley	1980/07/19	2008/03/16	2	11000
3	Rachel	1970/12/17	2010/12/01	7	9000

12.4.1 排序

【例12.11】按员工工资升序顺序取出员工姓名。员工表从文件EmployeeSample.txt中读取：

解题思路

先取得员工按工资按升序排列后的原序号，再按照原序号从员工表中取得姓名。

ID	Name	Salary
1	Rebecca	7000
2	Ashley	11000
3	Rachel	9000

	A	B
1	=file("EmployeeSample.txt").import@t()	/从文本文件中导入员工表
2	=A1.psort(Salary)	/使用psort()函数计算按工资升序排名的原序号
3	=A1.(Name)(A2)	/从员工表中，按工资排名序号取出姓名。

A3	Name
	Rebecca
	Rachel
	Ashley

注意

- psort() 函数用来获取排序后的原序号，并不会改变序列原有的顺序。psort() 函数同样支持z选项，用于逆序。

12.4.1 排序

对齐是指将有序集合的成员按照指定的顺序进行排列的过程。

排序和对齐对比

- 排序只能将数据按照升序或者降序排列。
- 如果需要将数据按照指定的顺序排序，就要使用对齐。

按Monday，Tuesday，Wednesday，Thursday，Friday的顺序查询工作日志。

Weekday	DailyRecord
Wednesday	Log1
Monday	Log2
Friday	Log3
Tuesday	Log4
Thursday	Log5

Weekday	DailyRecord
Monday	Log2
Tuesday	Log4
Wednesday	Log1
Thursday	Log5
Friday	Log3

12.4.1 排序

【例12.12】按月份顺序（January，February，…，December）查询2019销售额，数据从文件MonthSales.txt中读取：

Year	Month	Amount
2019	December	3063.0
2019	November	3868.6
2019	October	2713.5
…	…	…

12.4.1 排序

【例12.12】按月份顺序（January，February，…，December）查询2019销售额，数据从文件MonthSales.txt中读取：

Year	Month	Amount
2019	December	3063.0
2019	November	3868.6
2019	October	2713.5
…	…	…

解题思路

因为要按固定顺序查询，就不能使用排序了，需要使用对齐函数。

	A	B
1	=file("MonthSales.txt").import@t()	/导入销售数据
2	=A1.select(Year==2019)	/选出2019年数据
3	[January,February,March,April,May,June,July,August,September,October,November,December]	/定义1到12月的常数序列
4	=A2.align@s(A3, Month)	/使用align@s()函数，将销售表的月份按照A3对齐

A4

Year	Month	Amount
2019	January	3730.0
2019	February	1444.8
2019	March	2018.6
…	…	…

注意

- P.align@s(A,y) 函数将序列P的y，按照序列A的顺序对齐。

12.5 习题

【习题 12.1】质数列[2,3,5,7,11,13,17,19]中，第3个和倒数第2个是多少?

12.5 习 题

【习题 12.1】质数列[2,3,5,7,11,13,17,19]中，第3个和倒数第2个是多少?

解题思路

访问有序集合的单个或多个成员，可以直接用序号访问。

	A	B
1	[2,3,5,7,11,13,17,19]	/定义常数序列
2	=A1(3)	/访问第3个成员，也可以用A1.m(3)
3	=A1.m(-2)	/访问倒数第2个成员
4	=A1.m(3,-2)	/同时访问第3个和倒数第2个成员

执行结果：

5

17

[5,17]

12.5 习 题

【习题 12.2】查询2014年哪一天完成了15万的销售计划。数据从文件Sales.txt中读取：

OrderID	Customer	SellerId	OrderDate	Amount
10400	EASTC	1	2014/01/01	3063.0
10401	HANAR	1	2014/01/01	3868.6
10402	ERNSH	8	2014/01/02	2713.5
10403	ERNSH	4	2014/01/03	1005.9
…	…	…	…	…

12.5 习 题

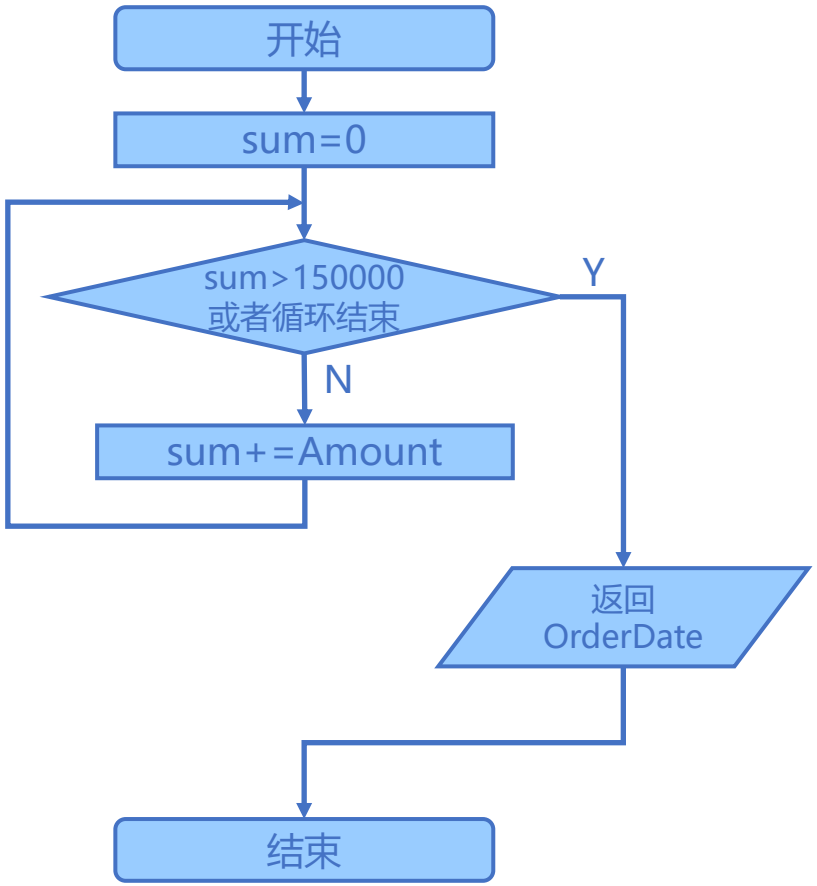
【习题 12.2】查询2014年哪一天完成了15万的销售计划。数据从文件Sales.txt中读取：

OrderID	Customer	SellerId	OrderDate	Amount
10400	EASTC	1	2014/01/01	3063.0
10401	HANAR	1	2014/01/01	3868.6
10402	ERNSH	8	2014/01/02	2713.5
10403	ERNSH	4	2014/01/03	1005.9
…	…	…	…	…

解题思路

思路是使用迭代计算每天的累计销售额，超过15万时终止迭代。

	A	B
1	=file("Sales.txt").import@t()	/导入销售表数据
2	=A1.select(year(OrderDate)==2014).sort(OrderDate)	/选出2014年数据并按销售日期升序排列
3	=sum=0, A2.iterate((sum+=Amount, ~~=OrderDate),0,sum>150000)	/定义sum变量代表累计销售额。循环累加销售额，累计超过15万时终止，并返回订单日期。



输出结果：2014/03/25

12.5 习 题

【习题 12.3】查询客户编号为AROUT的客户名称。数据从文件Customer.txt中读取：

ID	Name	City	Postcode	Tel	Fax
ALFKI	SanChuan	Tianjin	343567	(030) 30074321	(030) 30765452
ANATR	DongNan	Tianjin	234575	(030) 35554729	(030) 35553744
ANTON	TanShenHang	Shijiazhuang	985060	(0321) 5553932	(null)
AROUT	GuoDing	Shenzhen	890879	(0571) 45557788	(0571) 45556750
…	…	…	…	…	…

12.5 习 题

【习题 12.3】查询客户编号为AROUT的客户名称。数据从文件Customer.txt中读取：

ID	Name	City	Postcode	Tel	Fax
ALFKI	SanChuan	Tianjin	343567	(030) 30074321	(030) 30765452
ANATR	DongNan	Tianjin	234575	(030) 35554729	(030) 35553744
ANTON	TanShenHang	Shijiazhuang	985060	(0321) 5553932	(null)
AROUT	Guoding Co., Ltd	Shenzhen	890879	(0571) 45557788	(0571) 45556750
…	…	…	…	…	…

解题思路

思路是定位成员，找到客户ID是AROUT的序号，再访问其名称。

	A	B
1	=file("Customer.txt").import@t()	/导入客户表数据
2	=A1.(ID).pos("AROUT")	/使用pos()函数，定位客户编号。
3	=A1(A2).Name	/按序号访问客户表成员，并取出姓名

输出结果： Guoding Co., Ltd

注意

- A1(A2)的结果是记录。可以用记录.字段的方式访问记录的字段值。

12.5 习 题

【习题 12.3】查询客户编号为AROUT的客户名称。数据从文件Customer.txt中读取：

ID	Name	City	Postcode	Tel	Fax
ALFKI	SanChuan	Tianjin	343567	(030) 30074321	(030) 30765452
ANATR	DongNan	Tianjin	234575	(030) 35554729	(030) 35553744
ANTON	TanShenHang	Shijiazhuang	985060	(0321) 5553932	(null)
AROUT	Guoding Co., Ltd	Shenzhen	890879	(0571) 45557788	(0571) 45556750
…	…	…	…	…	…

解题思路

另外一种思路是选出客户ID是AROUT的成员，再访问其名称。

	A	B
1	=file("Customer.txt").import@t()	/导入客户表数据
2	=A1.select@1(ID:"AROUT")	/使用select@1()函数，选出第一个满足条件的客户
3	=A2.Name	/按序号访问客户表成员，并取出姓名

输出结果： Guoding Co., Ltd

注意

- 使用select@1()时，选出的结果是记录。

12.5 习 题

【习题 12.4】求2019年上证指数收盘价比开盘价增长最高的交易日，交易量较前日的涨幅。数据从文件SSEC.txt中读取：

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
…	…	…	…

12.5 习 题

【习题 12.4】求2019年上证指数收盘价比开盘价增长最高的交易日，交易量较前日的涨幅。数据从文件SSEC.txt中读取：

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
…	…	…	…

解题思路

有序集合按照指定条件（收盘价 - 开盘价）的最大值查找所在序号。再根据序号访问当日与前一日的交易量，计算涨幅。

	A	B
1	=file("SSEC.txt").import@t()	/导入上证指数数据
2	=A1.select(year(Date)==2019).sort(Date)	/选出2019年数据，并按日期升序排列
3	=A2.pmax(Close-Open)	/使用pmax()函数，定位收盘价比开盘价增长最高的序号。
4	=if(A3>1, A2(A3).Amount/A2(A3-1).Amount, 1.0)	/按序号取出最高增长当日和前日的交易量，并计算涨幅。

A3
34

A4
1.75849

12.5 习 题

【习题 12.5】求2019年上证指数收盘价涨幅超过3%的交易日，交易量较前日的涨幅。数据从文件SSEC.txt中读取：

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
…	…	…	…

12.5 习 题

【习题 12.5】求2019年上证指数收盘价涨幅超过3%的交易日，交易量较前日的涨幅。数据从文件SSEC.txt中读取：

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
…	…	…	…

解题思路

有序集合按照指定条件查找序号，再根据序号访问当日与前一日的交易量，计算涨幅。

	A	B
1	=file("SSEC.txt").import@t()	/导入上证指数数据
2	=A1.select(year(Date)==2019).sort(Date)	/选出2019年数据，并按日期升序排列
3	=A2.pselect@a(Close/Close[-1]>1.03)	/使用pselect@a()函数，定位所有收盘价与前日相比涨幅超过1.03的序号。
4	=A3.new(A2(~).Date:Date, if(~>1, A2(~).Amount/A2(~-1).Amount, 1.0):Increase)	/创建序表。循环定位的序号，计算这几个交易日与前日的交易量涨幅。

A3	Member
	34
	58
	84

A4	Date	Increase
	2019/02/25	1.75849
	2019/03/29	1.33448
	2019/05/10	1.39086

12.5 习 题

【习题 12.6】查询历届奥运会的奖牌榜，按赛事和名次进行降序排列。数据从文件Olympic.txt中读取：

Game	Nation	Gold	Silver	Copper
1	USA	11	7	1
1	Greece	10	19	17
1	Germany	7	5	2
1	France	5	4	2
…	…	…	…	…

12.5 习 题

【习题 12.6】查询历届奥运会的奖牌榜，按赛事和名次进行降序排列。数据从文件Olympic.txt中读取：

Game	Nation	Gold	Silver	Copper
1	USA	11	7	1
1	Greece	10	19	17
1	Germany	7	5	2
1	France	5	4	2
…	…	…	…	…

解题思路

可以同时按第几届和名次降序排列，其中名次顺序需要依次按照金银铜牌数量比较。

	A	B
1	=file("Olympic.txt").import@t()	/导入奥运奖牌榜数据
2	=A1.sort@z(Game, 1000000*Gold+1000*Silver+Copper)	/使用sort()函数按第几届和总成绩排序，z选项表示逆序

A2	Game	Nation	Gold	Silver	Copper
	30	USA	46	29	29
	30	China	38	27	23
	30	UK	29	17	19
	…	…	…	…	…

【习题 12.7】按单价升序顺序列出产品名称。产品表数据从文件ProdctPrice.txt中读取：

ID	Name	SUP_ID	Category	Price
1	Apple	2	1	18.5
2	Milk	1	1	21.0
3	Tomato	1	2	12.0
4	Salt	2	2	15.0

12.5 习 题

【习题 12.7】按单价升序顺序列出产品名称。产品表数据从文件ProdctPrice.txt中读取：

ID	Name	Price
1	Apple	18.5
2	Milk	21.0
3	Tomato	12.0
4	Salt	15.0

解题思路

先取得按单价升序排列后的序号，再按照序号从序表中取得姓名。

	A	B
1	=file("ProdctPrice.txt").import@t()	/导入产品表数据
2	=A1.psort@z(Price)	/使用psort()函数的@z选项获取单价的降序序号
3	=A1.(Name)(A2)	/从产品表中，按单价序号取出姓名。

A3	NAME
	Milk
	Apple
	Salt
	Tomato

第13章 分组计算

13.1 分组汇总

- 分组汇总
- 计算唯一值

13.2 分组的本质

- 分组子集
- 分组汇总与分组

13.3 非等值分组

- 等值分组与非等值分组
- 对齐分组
- 枚举分组

13.4 有序分组

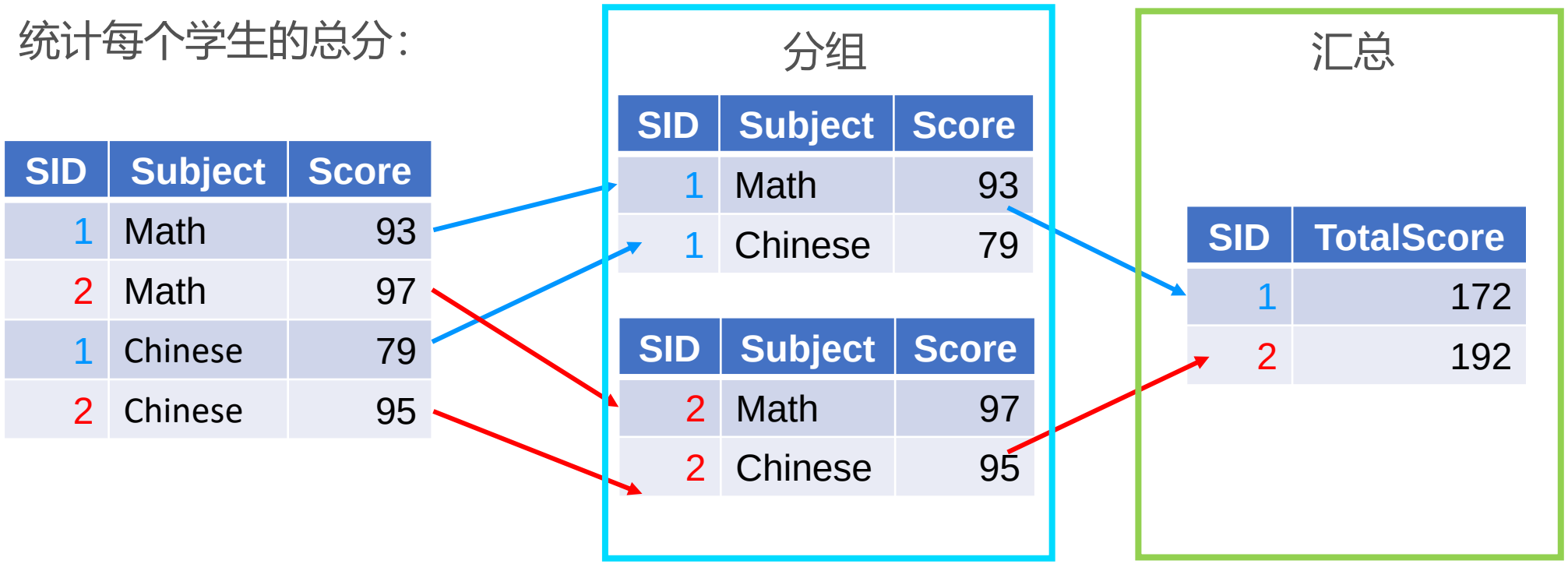
- 分组的顺序
- 有序分组

13.5 习题

13.1 分组汇总

13.1.1 分组汇总

有时候我们需要将数据按不同类型进行统计，这就要用到分组汇总。分组汇总是指根据一定的规则将数据进行分组，然后针对每个分组进行聚合运算。



13.1.1 分组汇总

【例13.1】统计2014年每个客户的总销售额。数据从文件 Sales.txt 中读取。

OrderID	Customer	OrderDate	Amount
10400	EASTC	2014/01/01	3063.0
10401	RATTC	2014/01/01	3868.6
10402	ERNSH	2014/01/02	2713.5
…	…	…	…

13.1.1 分组汇总

【例13.1】统计2014年每个客户的总销售额。数据从文件 Sales.txt 中读取。

解题思路

SPL 的 **groups** 函数可以实现**分组汇总**。函数 **A.groups(x:F,...;y:G,...)** 将序表按照一个或多个字段/表达式进行分组聚合后，形成以F,... G,...为字段的新序表。其中 **y** 是聚合函数。

	A	B
1	=file("Sales.txt").import@t() .select(year(OrderDate)==2014)	/从文件中导入销售表，并选出2014年的销售记录
2	=A1.groups(Customer; sum(Amount):Amount)	/分组汇总每个客户的总销售额

A2	Customer	Amount
	ANATR	1129.75
	…	…

注意

- 函数 **sum()** 用来**求和**，可以用于 **groups** 函数中进行聚合运算。

13.1.1 分组汇总

【例13.2】统计各部门员工的最高工资、最低工资和平均年龄。数据从文件 Employee.txt 中读取。

ID	Name	Department	Birthday	Salary
1	Rebecca	6	1974/11/20	7000
2	Ashley	2	1980/07/19	11000
3	Rachel	7	1970/12/17	9000
…	…	…	…	…

13.1.1 分组汇总

【例13.2】统计各部门员工的最高工资、最低工资和平均年龄。数据从文件 Employee.txt 中读取。

解题思路

对员工表按照部门分组，并统计每个部门的最高工资、最低工资和平均年龄。**分组汇总**可以使用 **A.groups()** 函数，聚合函数和聚合字段可以是多个。

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=A1.groups(Department; max(Salary):MaxSalary, min(Salary):MinSalary, round(avg(age(Birthday)),1):AvgAge)	/分组汇总各部门的最高工资、最低工资和平均年龄

A2	Department	MaxSalary	MinSalary	AvgAge
	1	16000	5000	44.0
	2	11000	5000	39.3
	3	12000	3000	41.4
	…	…	…	…

注意

- 函数 **avg()** 用来计算**平均值**，函数 **max()** 用来取**最大值**，函数 **min()** 用来取**最小值**。都可以用于 **groups** 函数中进行聚合运算。

13.1.1 分组汇总

【例13.3】统计每个班各科优秀（超过85分）的人数和分数的中位数。数据从文件Score.txt中读取。

CLASS	STUDENTID	SUBJECT	SCORE
1	1	English	84
1	1	Math	77
1	1	Chinese	69
2	1	English	96
2	1	Math	63
2	1	Chinese	81
1	2	English	81
1	2	Math	80
1	2	Chinese	97
…	…	…	…

13.1.1 分组汇总

【例13.3】统计每个班各科优秀（超过85分）的人数和分数的中位数。数据从文件Score2.txt中读取。

解题思路

对成绩表按照班级和科目两个字段分组，并汇总优秀人数和分数的中位数。**分组汇总**可以使用**A.groups()** 函数，其中分组字段/表达式、聚合函数都可以是多个。

	A	B
1	=file("Score2.txt").import@t()	/从文件中导入成绩表
2	=A1.groups(CLASS,SUBJECT; count(SCORE>85):ExcellentCount, median(,SCORE):Median)	/按班级和科目分组，汇总优秀人数和计算分数的中位数

A2

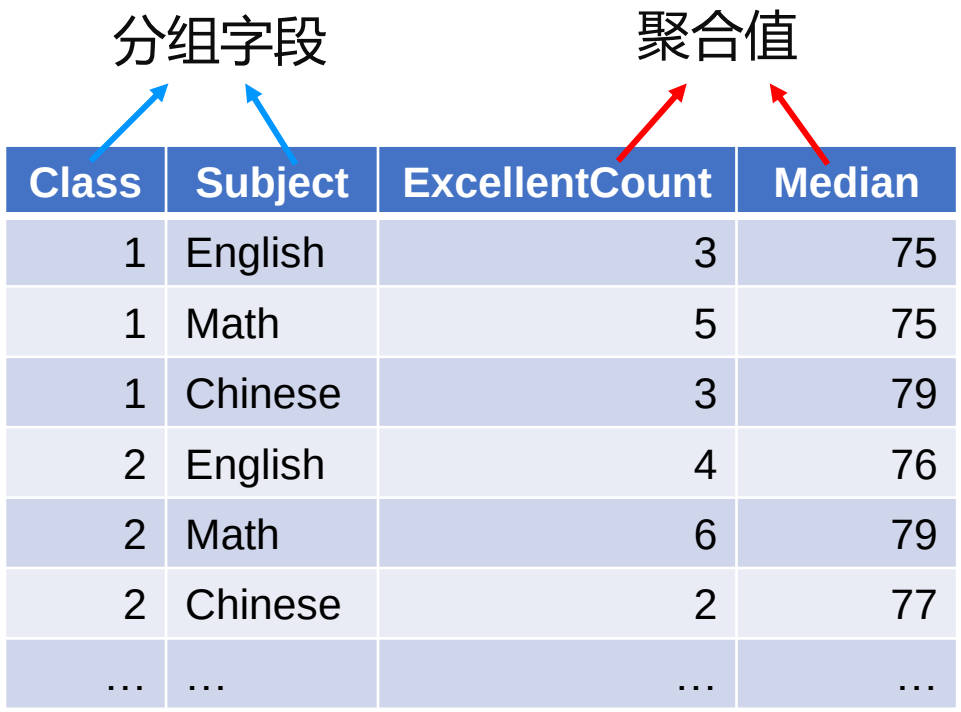
CLASS	SUBJECT	ExcellentCount	Median
1	Chinese	3	79
1	English	3	75
1	Math	5	75
…	…	…	…

注意

- 函数 **count()** 用来**计数**，函数 **median(, x)** 用来计算**中位数**。都可以用于 **groups** 函数中进行聚合运算。

13.1.1 分组汇总

分组汇总运算有两个关键因素：分组字段和聚合值。
分组汇总运算的结果是一个由分组字段和聚合值构成的序表



Class	Subject	ExcellentCount	Median
1	English	3	75
1	Math	5	75
1	Chinese	3	79
2	English	4	76
2	Math	6	79
2	Chinese	2	77
…	…	…	…

在分组汇总结果中，**分组字段**（可能多个）**是唯一的**。

每个 Class 和 Subject 确定了**唯一**一条记录。

13.1.1 分组汇总

【例13.4】统计2014年，客户排进月销售额前三的次数。数据从文件Sales.txt中读取。

OrderID	Customer	OrderDate	Amount
10400	EASTC	2014/01/01	3063.0
10401	RATTC	2014/01/01	3868.6
10402	ERNSH	2014/01/02	2713.5
…	…	…	…

13.1.1 分组汇总

【例13.4】统计2014年，客户排进月销售额前三的次数。数据从文件Sales.txt中读取。

解题思路

对销售表按照订购日期的月份分组，并统计每个月的总销售额。**分组汇总**可以使用 **A.groups()** 函数。

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=A1.select(year(OrderDate)==2014)	/从销售表中选出2014年的记录
3	=A2.groups(month(OrderDate):Month; top(-3;Amount):Top3)	/分组汇总每个月的销售额前三名
4	=A3.conj(Top3).groups(Customer; count(~):Count)	/再次分组汇总每个客户出现的次数

注意

- 函数 **top(n; x)** 用来取**x值的前n名/后n名**，当 **n** 为正数时取 **x** 的后 **n** 名，当 **n** 为负数时取 **x** 的前 **|n|** 名。可以用于 **groups** 函数中进行聚合运算。

13.1.1 分组汇总

【例13.4】统计2014年，客户排进月销售额前三的次数。数据从文件Sales.txt中读取。

A3	Month	Top3
	1	[[10424,MEREP,2014/01/23,11493.2], …]
	2	[[10440,HANAR,2014/02/10,10793.1], …]
	3	[[10479,HANAR,2014/03/19,10495.6], …]
	…	…

OrderID	Customer	OrderDate	Amount
10424	MEREP	2014/01/23	11493.2
10417	SIMOB	2014/01/16	11283.2
10430	ERNSH	2014/01/30	5796.0

分组字段

A4	Customer	Count
	ERNSH	4
	FOLIG	1
	FOLKO	1
	…	…

在分组汇总结果序表中，**分组字段的值**具备**唯一性**。A3中没有重复的月份，A4中不会有重复的客户。

13.1.1 分组汇总

以【例13.1】为例，看一下 groups 函数的实质计算过程。统计2014年每个客户的总销售额。数据从文件Sales.txt中读取。

OrderID	Customer	OrderDate	Amount
10400	EASTC	2014/01/01	3063.0
10401	RATTC	2014/01/01	3868.6
10402	ERNSH	2014/01/02	2713.5
…	…	…	…

13.1.1 分组汇总

【例13.1】统计2014年每个客户的总销售额。数据从文件 Sales.txt 中读取。

解题思路

函数 groups 在**分组汇总**时用累计方式计算。首先创建一个空序表，用于存放汇总结果。循环数据表的每条记录，在汇总结果中按分组字段值进行查找。找不到时创建新组，找到时进行累计运算。最后再按照分组字段进行排序。

Customer	Amount
EASTC	3063.0
RATTC	3868.6
…	…
RATTC	3674.0
…	…

Customer	Amount

创建一个空序表，用于存放汇总结果。

Customer	Amount
EASTC	3063.0

找不到EASTC时创建新组

Customer	Amount
EASTC	3063.0
RATTC	3868.6

找不到RATTC时创建新组

Customer	Amount
EASTC	3063.0
RATTC	7,542.6

找到RATTC时累加销售额

13.1.1 分组汇总

【例13.1】统计2014年每个客户的总销售额。数据从文件 Sales.txt 中读取。

解题思路

可以用循环函数模拟函数 **groups** 的计算原理（以 **sum** 为例），SPL 如下：

	A	B	C	D
1	=file("Sales.txt").import@t().select(year(OrderDate)==2014)			/从文件中导入销售表，并选出2014年的销售记录
2	=create(Customer, Amount).keys(Customer)			/创建空序表，并设置客户为主键
3	for A1	=A2.find(A3.Customer)		/循环销售表，在汇总结果中查找当前客户
4		if B3==null	>A2.insert(0, A3.Customer, A3.Amount)	/找不到时创建新组
5		else	>B3.Amount+=A3.Amount	/找到时累加
6	=A2.sort(Customer)			/按客户排序

A6	Customer	Amount
	ANATR	1129.75
	…	…

13.1.2 计算唯一值

前面介绍过，分组汇总的结果集，是由**分组字段**和**聚合值**构成的序表。在结果序表中，**分组字段的值具有唯一性**，所以**字段值去重**可以通过没有聚合值的分组来实现。

统计销售了哪些产品：

Order	Product
1	Coffee
2	Milk
~~3~~	~~Coffee~~
~~4~~	~~Milk~~

计算唯一值

Coffee

Milk

重复值仅保留一个

13.1.2 计算唯一值

【例13.5】查询员工所在的州和所属部门有哪些。数据从文件Employee.txt中读取。

ID	Name	State	Department
1	Rebecca	California	6
2	Ashley	New York	2
3	Rachel	New Mexico	7
…	…	…	…

13.1.2 计算唯一值

【例13.5】查询员工所在的州和所属部门有哪些。数据从文件Employee.txt中读取。

解题思路

前面讲过没有聚合值时的分组可以得到**唯一值**。函数 **A.groups(x:F,…)** 可以用来计算**唯一值**。

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=A1.groups(State,Department)	/使用groups函数计算唯一值

A2	State	Department
	Alabama	4
	Alabama	7
	Arizona	5
	Arizona	7
	Arizona	8
	Arkansas	6
	…	…

注意

- 函数 **groups** 计算唯一值的结果集是**分组字段/表达式**组成的**序表**，与分组汇总的结果集类似，只是没有聚合值。

13.1.2 计算唯一值

【例13.6】查询2014年有销售记录的客户。数据从文件Sales.txt中读取。

OrderID	Customer	OrderDate	Amount
10400	ANATR	2014/01/01	3063.0
10401	ERNSH	2014/01/01	3868.6
10402	ANATR	2014/01/02	2713.5
…	…	…	…

13.1.2 计算唯一值

【例13.6】查询2014年有销售记录的客户。数据从文件Sales.txt中读取。

解题思路

取单字段的**唯一值**问题。可以使用 **A.groups(x)** 函数取 **x** 的**唯一值**。

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=A1.select(year(OrderDate)==2014)	/选出2014年的销售记录
3	=A2.groups(Customer)	/使用groups函数计算客户的唯一值

A3	Customer
	ANATR
	ANTON
	AROUT
	…

注意

- 函数 **groups** 返回的结果总是**序表**。计算单字段**唯一值**时，返回的是单字段的**序表**，而不是**序列**。

13.1.2 计算唯一值

【例13.6】查询2014年有销售记录的客户。数据从文件Sales.txt中读取。

解题思路

计算**唯一值**的分组字段经常只有一个，**groups** 返回一个单字段**序表**不方便，所以简化成 **id** 了。SPL 的 **A.id(x_i, ...)** 函数，用于取每个 **x_i** 的**唯一值**。

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=A1.select(year(OrderDate)==2014)	/选出2014年的销售记录
3	=A2.id(Customer)	/使用id函数计算客户的唯一值

A3	Member
	ANATR
	ANTON
	AROUT
	…

注意

- 函数 **A.id(x_i, ...)** 只有一个 **x_i** 时返回成一个**序列**，多个 **x_i** 时返回成**序列构成的序列**。

13.1.2 计算唯一值

以【例13.5】为例看一下多字段时 id 函数的返回值。查询员工所在的州和所属部门有哪些。数据从文件Employee.txt中读取。

解题思路

函数 A.id(x_i, ...) 当 x_i ... 是多字段时，返回由唯一值序列组成的序列。

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=A1.id(State,Department)	/使用id函数计算唯一值

A2	Member
	[California,New York,New Mexico,...]
	[6,2,7,3,4,5,1,8]

Member	Member
California	6
New York	2
…	…

13.1.2 计算唯一值

【例13.7】统计2014年每个月有多少个客户。数据从文件Sales.txt中读取。

OrderID	Customer	OrderDate	Amount
10400	ANATR	2014/01/01	3063.0
10401	ERNSH	2014/01/01	3868.6
10402	ANATR	2014/01/02	2713.5
…	…	…	…

13.1.2 计算唯一值

【例13.7】统计2014年每个月有多少个客户。数据从文件Sales.txt中读取。

解题思路

对销售表按照月份分组，并统计客户的**唯一值的个数。分组汇总**可以使用 **A.groups()** 函数。

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=A1.select(year(OrderDate)==2014)	/选出2014年的销售记录
3	=A2.groups(month(OrderDate):Month; icount(Customer):Count)	/分组汇总每个月的客户数量

A3	Month	Count
	1	27
	2	22
	3	24
	…	…

注意

- 函数 **icount()** 用来统计表达式**唯一值的个数**，可以用于 **groups** 函数中进行聚合运算。

13.2 分组的本质

13.2.1 分组子集

分组运算的实质是将一个集合按照某种规则拆分成若干个子集，也就是说，返回值应当是一个由集合构成的集合。对于每个成员集合，我们称为**分组子集**。

统计每个学生的总分：

SID	Subject	Score
1	Math	93
2	Math	97
1	Chinese	79
2	Chinese	95

分组子集

SID	Subject	Score
1	Math	93
1	Chinese	79

SID	Subject	Score
2	Math	97
2	Chinese	95

13.2.1 分组子集

通常人们对分组子集的聚合值更感兴趣，因此分组运算常常伴随着对子集的进一步汇总计算。但是，我们仍然有对这些分组子集而不是聚合值更感兴趣的时候。

查看偏科学生（两科分数差10分以上）的各科成绩：

SID	Subject	Score
1	Math	93
2	Math	97
1	Chinese	79
2	Chinese	95

分组

Member
[[1,Math,93], [1,Chinese,79]]
[[2,Math,97], [2,Chinese,95]]

满足条件的分组子集

SID	Subject	Score
1	Math	93
1	Chinese	79

分组汇总只能得到聚合结果，无法获取**分组子集**中的详细信息。

13.2.1 分组子集

【例13.8】找出有相同出生日期的员工。数据从文件Employee.txt中读取。

ID	Name	Birthday	Salary
1	Rebecca	1974/11/20	7000
2	Ashley	1980/07/19	11000
3	Rachel	1970/12/17	9000
…	…	…	…

13.2.1 分组子集

【例13.8】找出有相同出生日期的员工。数据从文件Employee.txt中读取。

解题思路

这里我们关心的并不是聚合结果，而是**分组子集**，就需要使用**分组**了。函数 **A.group(x_i, ...)** 用于按x_i,...做等值分组并排序，返回值是序列（**分组子集**）的序列。

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=A1.group(Birthday)	/使用group函数按出生日期分组，不聚合
3	=A2.select(~.len()>1).conj()	/选出有相同出生日期的员工

A2

Member
[296,Olivia,1968/11/05,10000]
[[67,Cameron,1970/03/14,5000],[115,Gabriel,1970/03/14,7000]]
[32,Andrew,1971/08/27,11000]
…

ID	Name	Birthday	Salary
296	Olivia	1968/11/05	10000

ID	Name	Birthday	Salary
67	Cameron	1970/03/14	5000
115	Gabriel	1970/03/14	7000

A3

ID	Name	Birthday	Salary
67	Cameron	1970/03/14	5000
115	Gabriel	1970/03/14	7000
32	Andrew	1971/08/27	5000
47	Elizabeth	1971/08/27	6500
…	…	…	…

13.2.1 分组子集

【例13.9】计算每个部门的人数，再计算出 50 人以上部门的人员平均年龄。数据从文件Employee.txt中读取。

ID	Name	Birthday	Department
1	Rebecca	1974/11/20	6
2	Ashley	1980/07/19	2
3	Rachel	1970/12/17	7
…	…	…	…

13.2.1 分组子集

【例13.9】计算每个部门的人数，再计算出 50 人以上部门的人员平均年龄。数据从文件Employee.txt中读取。

解题思路

分组本身是复杂运算，保留**分组子集**可重复使用，提高运行效率。第二次汇总时可以利用第一次的**分组子集**继续计算，不需要再次**分组**。

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=A1.group (Department)	/使用group函数按部门分组
3	=A2.new(Department, count(~):Count)	/汇总计算每个部门的人数
4	=A2.select(~.count()>50).new(Department, ~.avg(age(Birthday)):Age)	/选出人数大于50的部门子集，统计平均年龄

A2	Member
	[[18,Jonathan,1971/03/07,1],…]
	…

ID	Name	Birthday	Department
18	Jonathan	1971/03/07	1
…	…	…	…

A3	Department	Count
	1	4
	2	24
	…	…

A4	Department	Age
	4	42.56
	5	41.33
	7	41.95

13.2.1 分组子集

以【例13.8】为例，看一下 group 函数的分组过程。找出有相同出生日期的员工。数据从文件Employee.txt中读取。

ID	Name	Birthday	Salary
1	Rebecca	1974/11/20	7000
2	Ashley	1980/07/19	11000
3	Rachel	1970/12/17	9000
…	…	…	…

13.2.1 分组子集

【例13.8】找出有相同出生日期的员工。数据从文件Employee.txt中读取。

解题思路

函数 **group** 通过分组字段/表达式进行**分组**。首先创建一个空序列，用于存放分组结果。循环数据表的每条记录，在分组结果中按分组字段值进行查找。找不到时创建新组，找到时在定位的组追加记录。最后再按照分组字段进行排序。

ID	Birthday
1	1974/11/20
…	…
67	1970/03/14
…	…
115	1970/03/14
…	…

创建一个空序表，用于存放汇总结果。

找不到1974/11/20时创建新组

找不到1970/03/14时创建新组

找到1970/03/14时在该组追加记录

ID	Birthday
67	1970/03/14
115	1970/03/14

13.2.1 分组子集

【例13.8】找出有相同出生日期的员工。数据从文件Employee.txt中读取。

解题思路

可以用循环函数模拟函数 **group** 的**分组**计算原理，SPL 如下：

	A	B	C	D	E
1	=file("Employee.txt").import@t()				/从文件中导入员工表
2	[]				/创建空序表存储分组结果
3	for A1	if A2.len()==0	>A2=[[A3]]		/循环员工表，第一条插入空序列
4		else	=A2.(Birthday).pos(A3.Birthday)		/在结果集中定位出生日期
5			if C4==null	>A2=A2|[[A3]]	/找不到时创建新组
6			else	>A2(C4)=A2(C4)|A3	/找到时在定位的组追加
7	=A2.sort(Birthday)				/按出生日期排序
8	=A7.select(~.len()>1).conj()				/选出有相同出生日期的员工

13.2.1 分组子集

【例13.8】找出有相同出生日期的员工。数据从文件Employee.txt中读取。

A2

Member
[296,Olivia,1968/11/05,10000]
[[67,Cameron,1970/03/14,5000],[115,Gabriel,1970/03/14,7000]]
[32,Andrew,1971/08/27,11000]
…

ID	Name	Birthday	Salary
296	Olivia	1968/11/05	10000

ID	Name	Birthday	Salary
67	Cameron	1970/03/14	5000
115	Gabriel	1970/03/14	7000

A8

ID	Name	Birthday	Salary
67	Cameron	1970/03/14	5000
115	Gabriel	1970/03/14	7000
32	Andrew	1971/08/27	5000
47	Elizabeth	1971/08/27	6500
…	…	…	…

13.2.2 分组汇总与分组

【例13.10】统计每个班每个学生的总分数和平均分数。数据从文件Score.txt中读取。

CLASS	STUDENTID	SUBJECT	SCORE
1	1	English	84
1	1	Math	77
1	1	Chinese	69
2	1	English	96
2	1	Math	63
2	1	Chinese	81
1	2	English	81
1	2	Math	80
1	2	Chinese	97
…	…	…	…

13.2.2 分组汇总与分组

【例13.10】统计每个班每个学生的总分数和平均分数。数据从文件Score.txt中读取。

解题思路

先按班级和学生进行**分组**，再对**分组子集**进行聚合计算。

	A	B
1	=file("Score.txt").import@t()	/从文件中导入成绩表
2	=A1.group(CLASS, STUDENTID)	/使用group函数按班级和学生分组
3	=A2.new(CLASS, STUDENTID, ~.sum(SCORE):TotalScore, round(~.avg(SCORE), 2):AvgScore)	/统计每个学生的总分数和平均分数

A2

Member
[[1,1,English,84], …]
[[1,2,English,81], …]
[[1,3,English,75], …]
…

CLASS	STUDENTID	Subject	Score
1	1	English	84
1	1	Math	77
1	1	Chinese	69

A3

CLASS	STUDENTID	TotalScore	AvgScore
1	1	230	76.67
1	2	258	86.0
1	3	228	76.0
…	…	…	…

13.2.2 分组汇总与分组

【例13.10】统计每个班每个学生的总分数和平均分数。数据从文件Score.txt中读取。

解题思路

可以使用带聚合函数的 **group** 函数来简化。函数 **A.group(x:F,...;y:G,...)** 将序列 **A** 按照 **x,...** 先进行分组，然后再进行 **y** 运算。

	A	B
1	=file("Score.txt").import@t()	/从文件中导入成绩表
2	=A1.group(CLASS, STUDENTID; ~.sum(SCORE):TotalScore, round(~.avg(SCORE), 2):AvgScore)	/使用group函数按班级和学生分组，统计每个学生的总分数和平均分数。

A2	CLASS	STUDENTID	TotalScore	AvgScore
	1	1	230	76.67
	1	2	258	86.0
	1	3	228	76.0
	…	…	…	…

13.2.2 分组汇总与分组

【例13.10】统计每个班每个学生的总分数和平均分数。数据从文件Score.txt中读取。

解题思路

使用**分组汇总**函数 **A.groups(x:F,...;y:G,...)** 也可以解决问题。函数 **groups** 的表达式与 **group** 非常接近，区别是 **y** 仅限于几种累计计算的聚合表达式。前面我们学习过函数 groups 实际执行过程是累计的方式，不产生**分组子集**，效率更高。

	A	B
1	=file("Score.txt").import@t()	/从文件中导入成绩表
2	=A1.groups(CLASS, STUDENTID; sum(SCORE):TotalScore, avg(SCORE):AvgScore)	/使用groups函数按班级和学生分组，统计每个学生的总分数和平均分数。

A2	CLASS	STUDENTID	TotalScore	AvgScore
	1	1	230	76.67
	1	2	258	86
	1	3	228	76
	…	…	…	…

13.2.2 分组汇总与分组

【例13.11】查询工资超过部门平均工资的员工。数据从文件Employee.txt中读取。

ID	Name	Department	Salary
1	Rebecca	6	7000
2	Ashley	2	11000
3	Rachel	7	9000
…	…	…	…

13.2.2 分组汇总与分组

【例13.11】查询工资超过部门平均工资的员工。数据从文件Employee.txt中读取。

解题思路

本题直接用 **groups** 进行**分组汇总**无法解决，需要分组后对每个**分组子集**进行复杂计算。函数 **A.group(x:F,...;y:G,...)** 中 **y** 不限于**聚合函数**，可以使用任意的**序列**函数（针对分组子集进行计算）。

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=A1.group(Department; (a=avg(Salary), ~.select(Salary>a)):High)	/使用group函数按部门分组，统计每个部门的平均工资，并选出高于平均工资的员工。
3	=A2.conj(High)	/合并选出的员工

A3	ID	Name	Department	Salary
	20	Alexis	1	16000
	42	Michael	1	12000
	51	Madison	3	9000
	…	…	…	…

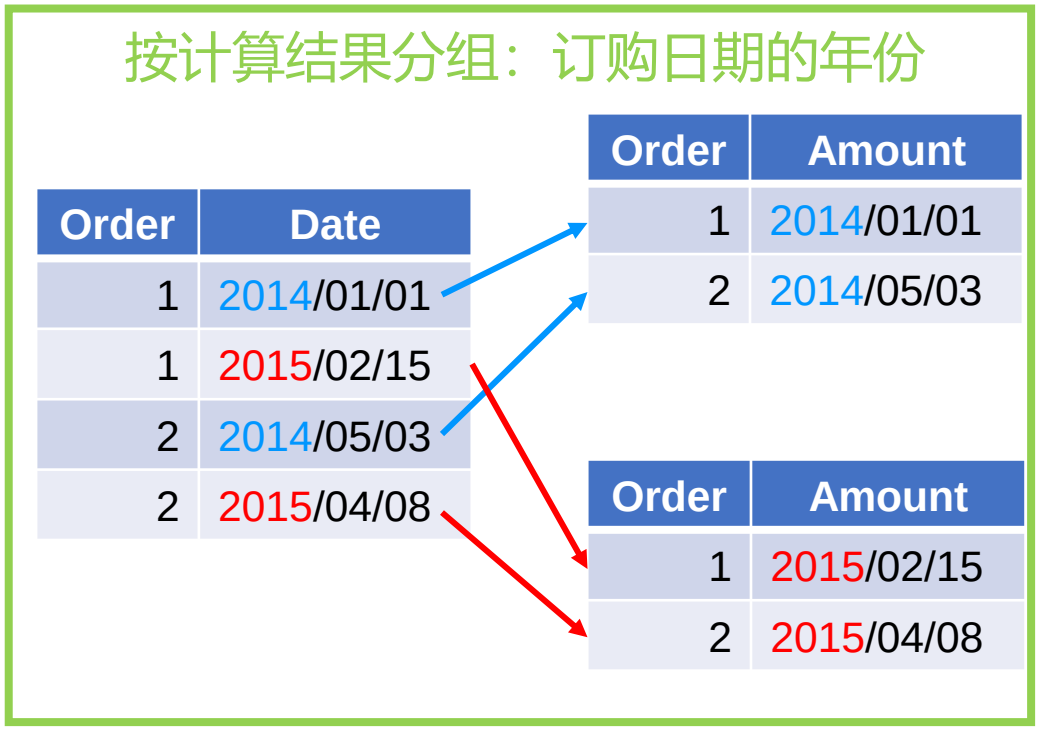
13.3 非等值分组

13.3.1 等值分组

我们之前碰到的分组问题都有一个特点，以相同的字段值或表达式结果为基准来分组，这种分组称为等值分组，是最常见的分组形式。等值分组具有完全划分的特点，即：

1.分组结果没有空子集。

2.原集合的任何成员都属于且只属于某一个子集。



13.3.1 非等值分组

我们学习了**等值分组**和**完全划分**。那是不是还有**非等值分组**和**不完全划分**？是的。还有具体如下特点的**不完全划分**：

1.分组结果包含空子集。
2.分组结果不包含所有原集合成员。
3.同一个成员分到两个组中。

按直辖市、一线城市、二线城市、三线城市分组：

1.没有成员分配到三线城市

2.分组结果不包含韶关（四线城市）

3.北京同时分配到直辖市和一线城市

ID	City
1	Beijing
2	Guangzhou
3	Dalian
4	Shaoguan

直辖市

ID	City
1	Beijing

一线城市

ID	City
1	Beijing
2	Guangzhou

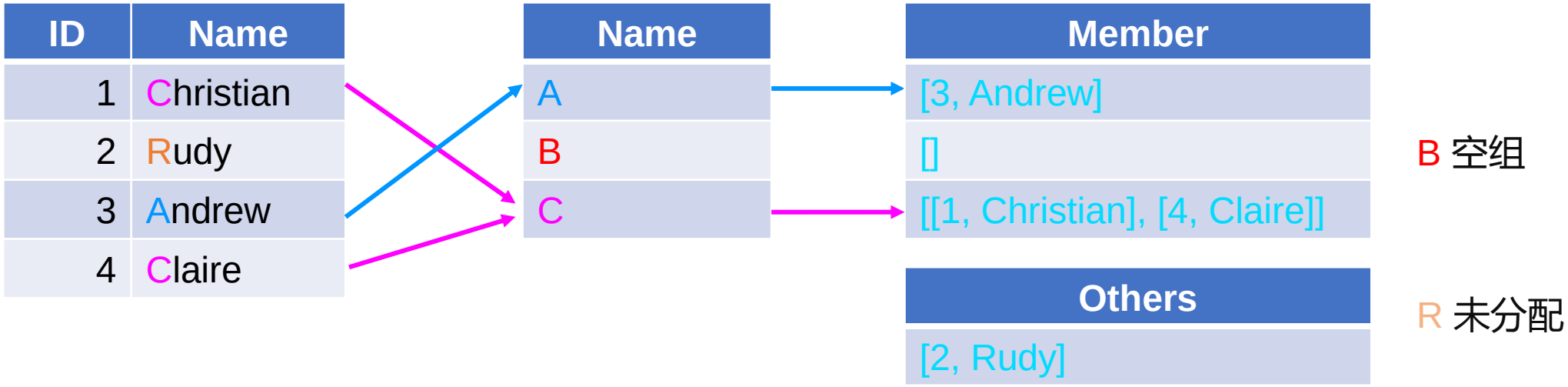
二线城市

ID	Name
3	Dalian

13.3.2 对齐分组

对照一个基准集合，将待分组集合成员的某个字段或表达式与基准集合成员比较，相同者则分到同一个组中，最后拆分出来的组数和基准集合成员数是相同的。这种分组我们称为对齐分组。对齐分组可能会有空组，也可能有成员未分配到任何一个组中。

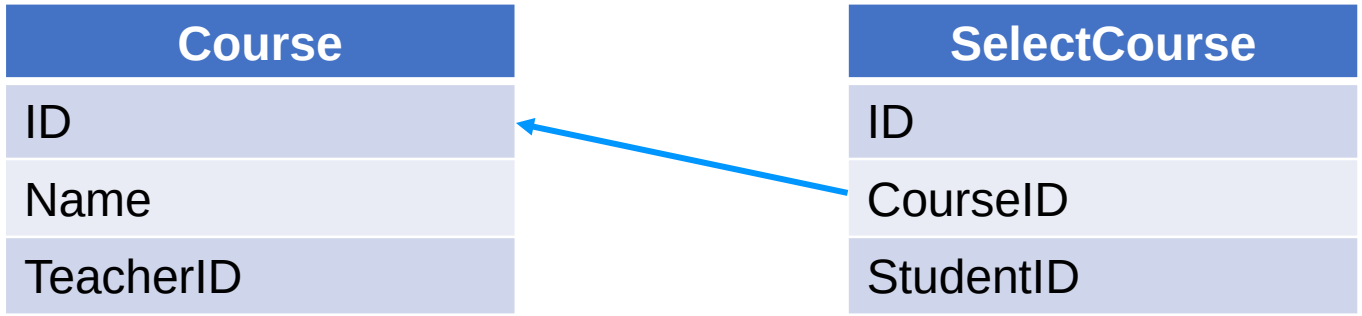
姓名的首字母按A、B、C顺序对齐分组：



13.3.2 对齐分组

每组选择一个匹配成员

【例13.12】按课程表顺序查询有哪些课没有学生选修。课程表和选课表数据分别从文件Course.txt 和 SelectCourse.txt 中读取。



13.3.2 对齐分组

每组选择一个匹配成员

【例13.12】按课程表顺序查询有哪些课没有学生选修。课程表和选课表数据分别从文件 Course.txt 和 SelectCourse.txt 中读取。

解题思路

选修记录按照课程表顺序分组，空组即为没有学生选修的课程，应该使用**对齐分组**。SPL 的 **P.align(A:x,y)** 函数将**P**的记录通过字段 **y** 向 **A** 记录的字段**x** 对齐。缺省每组选择一个匹配成员。

	A	B
1	=file("Course.txt").import@t()	/从文件中导入课程表
2	=file("SelectCourse.txt").import@t()	/从文件中导入选课表
3	=A2.align(A1:ID, CourseID)	/选修表按照课程表的序号对齐，每组选择一个匹配成员
4	=A1(A3.pos@a(null))	/从对齐分组结果中，选出空组，即没有学生选修的课程

A3	Member
	(null)
	[13,2,7]
	…
	[6,9,57]
	(null)

只关心是否空组，不在意每组成员，每组返回一个匹配成员即可

ID	CourseID	StudentID
13	2	7

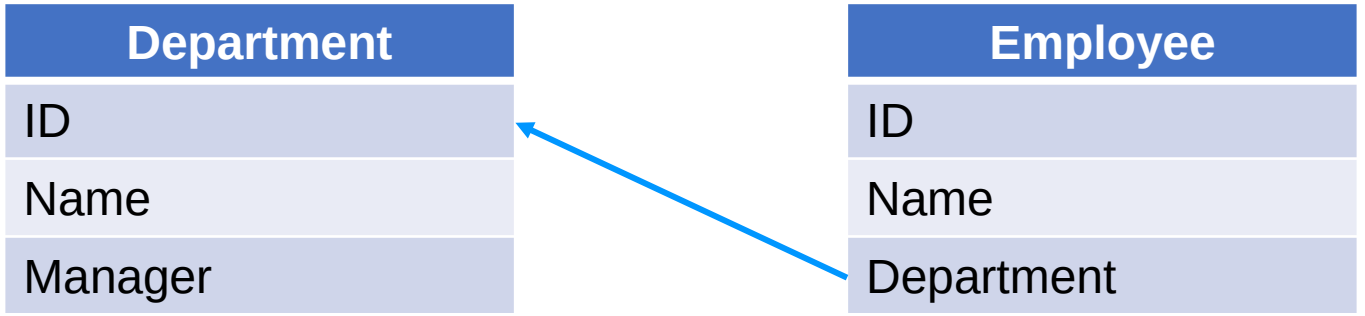
空组

A4	ID	NAME	TeacherID
	1	Environmental protection and sustainable development	5
	10	Music appreciation	18

13.3.2 对齐分组

每组选择所有匹配成员

【例13.13】按部门表的部门顺序统计各部门人数。部门表和员工表数据分别从文件Department.txt 和 Employee.txt 中读取。



13.3.2 对齐分组

【例13.13】按部门表的部门顺序统计各部门人数。部门表和员工表数据分别从文件Department.txt 和 Employee.txt 中读取。

解题思路

同样是**对齐分组**，与上一节不同的是，每组关心匹配成员的数量，这就需要选择所有匹配成员。
P.align (A:x,y) 函数的**@a** 选项可以返回每组匹配的所有成员。

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=file("Department.txt").import@t()	/从文件中导入部门表
3	=A1.align@a(A2:ID, Department)	/员工表按部门对齐分组，每组返回所有匹配成员
4	=A3.new(Department, ~.count():Count)	/统计各部门的员工数量

A3

Member
[[18,Jonathan,1,...], [20,Alexis, 1,...], …]
[[1,Rebecca,6,...],[5,Ashley,6,...],...]
[[3,Rachel,7,...],[6,Matthew,7,...],...]
…

每组返回所有匹配成员

Name	Department
Rebecca	6
Ashley	6
Ryan	6
…	…

A4

Department	Count
1	4
2	24
3	19
…	…

【例13.14】按周一到周五（Monday,Tuesday,Wednesday,Thursday,Friday）的顺序统计每个工作日的总销售额，周末的统计值存放到新组。数据从文件 DailySales.txt 中读取。

Week	Day	Amount
1	Sunday	2018.2
1	Monday	3063.0
1	Tuesday	3868.6
1	Wednesday	2713.5
1	Thursday	1005.9
1	Friday	1675.0
1	Saturday	400.0
2	Sunday	2123.2
…	…	…

13.3.2 对齐分组

【例13.14】按周一到周五（Monday,Tuesday,Wednesday,Thursday,Friday）的顺序统计每个工作日的总销售额，周末的统计值存放到新组。数据从文件 DailySales.txt 中读取。

解题思路

本例中除了每组需要选择所有匹配成员外，还要将不匹配的成员存放到新组。**P.align (A:x,y)** 函数的**@n** 选项会在基准集合最后创建一个新组，用于存放不匹配的成员。这里的基准集合是序列，参数x 可以省略。

	A	B
1	=file("DailySales.txt").import@t()	/从文件中导入销售表
2	[Monday,Tuesday,Wednesday,Thursday,Friday]	/创建周一到周五的常数序列
3	=A1.align@an(A2,Day)	/销售表按星期几对齐分组，每组返回所有匹配成员，不匹配成员存放到新组，即周末组。
4	=A3.new(if(#>A2.len(),"Weekend",Day):Weekday,~.sum(Amount):Amount)	/统计每组的总销售额，产生新序表。最后一组更名为Weekend，否则会显示为本组第一条记录的星期几。

13.3.2 对齐分组

不匹配成员存放到新组

【例13.14】按周一到周五（Monday,Tuesday,Wednesday,Thursday,Friday）的顺序统计每个工作日的总销售额，周末的统计值存放到新组。数据从文件 DailySales.txt 中读取。

A3	Member
	[[1, Monday, 3063.0], [2, Monday, 1194.0], …]
	[[1, Tuesday, 3868.6], [2, Tuesday, 1622.4], ...]
	[[1, Wednesday, 2713.5], [2, Wednesday, 319.2], ...]
	[[1, Thursday, 1005.9], [2, Thursday, 802.0], ...]
	[[1, Friday, 1675.0], [2, Friday, 1208.5], ...]
	[[1, Saturday, 400.0], [2, Sunday, 2018.2], …]

A4	Weekday	Amount
	Monday	8048.3
	Tuesday	9849.6
	Wednesday	4925.1
	Thursday	6458.5
	Friday	5491.5
	Weekend	15403.6

Week	Day	Amount
1	Sunday	2018.2
1	Saturday	400.0
2	Sunday	2123.2
2	Saturday	372.0
…	…	…

不匹配的成员存放到最后一组

13.3.3 枚举分组

枚举分组是指，事先指定一组**枚举条件**，将待分组集合的成员作为参数计算这批条件，条件成立者都被划分到与该条件对应的一个子集中，结果集的子集和事先指定的条件一一对应。

分组字段是Age：

ID	Name	Age
1	Rebecca	45
2	Ashley	39
3	Rachel	29
4	Emily	35

枚举条件

Subject
? < 30
? < 40

枚举分组

Member
[3,Rachel,29]
[[2,Ashley,39],[4,Emily,35]]

ID	Name	Age
2	Ashley	39
4	Emily	35

想一想：

（1）45没有满足的条件，未匹配的成员如何处理？

（2）29不仅满足Age<30的条件，同时也满足Age<40，是否重复分配到两个组？

13.3.3 枚举分组

每个成员只存放到第一个匹配组

【例13.15】下面是中国主要城市的市区人口表，根据人口将城市分类。数据从文件UrbanPopulation.txt 中读取。

ID	City	Population	Province
1	Shanghai	12286274	Shanghai
2	Beijing	9931140	Beijing
3	Chongqing	7421420	Chongqing
4	Guangzhou	7240465	Guangdong
5	Hong Kong	7010000	Hong Kong Special Administrative Region
…	…	…	…

超大城市200万人口以上，特大城市100-200万人口，大城市50-100万，其他中小城市。

13.3.3 枚举分组

每个成员只存放到第一个匹配组

【例13.15】下面是中国主要城市的市区人口表，根据人口将城市分类。数据从文件 UrbanPopulation.txt 中读取。

解题思路

根据人口与城市规模的关系创建条件常数序列。函数 **P.enum(E,y)** 依次对序列P中的每个成员计算表达式 **y**，将计算结果作为参数，可以让E中哪个条件表达式成立，就将这条记录分配到相应位置的组中。默认情况下，每个成员只存放到第一个匹配组。

ID	City	Population	Province
1	Shanghai	12286274	Shanghai
…	…	…	…

Member
?>2000000
?>1000000 && ?<=2000000
?>500000 && ?<=1000000
?<=500000

根据题目规则定义城市分组条件

还可以把条件简化，按每组的下限人口从大到小排列，每个成员只存放到第一个匹配组。

ID	City	Population	Province
1	Shanghai	12286274	Shanghai
…	…	…	…

Member
?>2000000
?>1000000
?>500000
?<=500000

上海虽然同时满足多个条件，但是只会分配到第一个匹配组，即超大城市组。

13.3.3 枚举分组

每个成员只存放到第一个匹配组

【例13.15】下面是中国主要城市的市区人口表，根据人口将城市分类。数据从文件UrbanPopulation.txt 中读取。

	A	B
1	=file("UrbanPopulation.txt").import@t()	/从文件中导入城市人口表
2	[?>2000000,?>1000000,?>500000,?<=500000]	/创建分组条件的常数序列
3	=A1.enum(A2, Population)	/市区人口表的人口按照A2的条件序列枚举分组，默认每个成员只存放到第一个匹配组

A3

Member
[[1,Shanghai,12286274,...], [2,Beijing, 9931140,...], …]
[[28,Changsha,1965282,...], [29,Nanchang,1900817,...], …]
[[69,Huainan,974026,...], [70,Haikou, 967336,...], …]
[]

ID	City	Population	Province
1	Shanghai	12286274	Shanghai
2	Beijing	9931140	Beijing
3	Chongqing	7421420	Chongqing

【例13.16】根据年龄区间：[35以下，35到49，其他] 将员工分组，并统计平均工资。数据从文件Employee.txt 中读取。

ID	Name	Birthday	Salary
1	Rebecca	1974/11/20	7000
2	Ashley	1980/07/19	11000
3	Rachel	1970/12/17	9000
…	…	…	…

13.3.3 枚举分组

不匹配成员存放到新组

【例13.16】根据年龄区间：[35以下，35到49，其他] 将员工分组，并统计平均工资。数据从文件Employee.txt 中读取。

解题思路

分组条件未必匹配所有的成员，这时可以将未匹配的成员存放到新组。函数 **P.enum(E,y)** 的 **@n** 选项，会在结果集最后创建一个新组，用于存放未匹配的成员。

ID	Name	Birthday	Salary
1	Rebecca	1974/11/20	7000
2	Ashley	1980/07/19	11000
3	Rachel	1970/12/17	9000
4	Emily	1986/03/07	7000
…	…	…	…

根据生日计算年龄

Age
46
40
50
34
…

Member
?<35
?<50

年龄按条件枚举分组

年龄50的员工不满足任何一个分组条件，分配到新组。

13.3.3 枚举分组

不匹配成员存放到新组

【例13.16】根据年龄区间：[35以下，35到49，其他] 将员工分组，并统计平均工资。数据从文件Employee.txt 中读取。

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	[?<35, ?<50]	/创建分组条件的常数序列
3	=A1.enum@n(A2, age(Birthday))	/首先根据生日计算出年龄。再按年龄枚举分组，不匹配成员存放到新组。
4	=A3.new(if(#>A2.len(),"Other",A2(#)):Age, round(~.avg(Salary), 2):AvgSalary)	/统计每组的平均工资，最后一组名称设置为Other

A3	Member
	[[64,Nathan,1986/09/29,5000], [73,Justin,1986/11/03,10000], …]
	[[1,Rebecca,1974/11/20,7000],[2,Ashley,1980/07/19,11000], …]
	[[67,Cameron,1970/03/14,5000],[74,Kayla,1969/07/06,5000], …]

ID	Name	Birthday	Salary
64	Nathan	1986/09/29	5000
37	Justin	1986/11/03	10000
…	…	…	…

A4	Age	AvgSalary
	?<35	7080.36
	?<50	7454.88
	Other	7191.18

13.3.3 枚举分组

【例13.17】下面是城市GDP表，分别统计直辖市、一线城市、二线城市的人均GDP。数据从文件GDP.txt 中读取。（GDP单位为亿元，Population单位为万人）

ID	City	GDP	Population
1	Shanghai	32679	2418
2	Beijing	30320	2171
3	Shenzhen	24691	1253
4	Guangzhou	23000	1450
5	Chongqing	20363	3372
…	…	…	…

13.3.3 枚举分组

【例13.17】下面是城市GDP表，分别统计直辖市、一线城市、二线城市的人均GDP。数据从文件GDP.txt 中读取。（GDP单位为亿元，Population单位为万人）

解题思路

需要注意的是，分组可能会有重复成员，比如北京既是一线城市，又是直辖市。函数 **P.enum(E,y)** 的**@r** 选项用于重复枚举，即将P的某条记录分在匹配的多组中。

统计直辖市和一线城市时，都应该统计北京，把北京重复划分到两个组。

ID	City	GDP	Population
…	…	…	…
2	Beijing	30320	2171
…	…	…	…

Member
直辖市
一线城市
二线城市

13.3.3 枚举分组

【例13.17】下面是城市GDP表，分别统计直辖市、一线城市、二线城市的人均GDP。数据从文件GDP.txt 中读取。

	A	B
1	=file("GDP.txt").import@t()	/从文件中导入GDP表
2	[Beijing,Shanghai,Tianjin,Chongqing]	/定义直辖市常数序列
3	[Beijing,Shanghai,Guangzhou,Shenzhen]	/定义一线城市常数序列
4	[Chengdu,Hangzhou,Chongqing,Wuhan,Xian,Suzhou,Tianjing,Nanjing,Changsha,Zhengzhou,Dongguan,Qingdao,Shenyang,Ningbo,Kunming]	/定义二线城市常数序列
5	[A2.pos(?)>0,A3.pos(?)>0, A4.pos(?)>0]	/定义枚举条件常数序列
6	=A1.enum@r(A5,City)	/按城市枚举分组
7	=A6.new(["Municipality","First-tier","Second-tier"](#):Area,~.sum(GDP)/~.sum(Population)*10000:CapitaGDP)	/统计每组的人均GDP

A6	Member
	[[1,Shanghai,32679,2418],[1,Beijing,30320,2171],...]
	[[1,Shanghai,32679,2418],[1,Beijing,30320,2171],...]
	[[5,Chongqing,20363,3372],[6,Tianjin,18809,1557],...]

ID	City	GDP	Population
1	Shanghai	32679	2418
…	…	…	…

A7	Area	CapitaGDP
	Municipality	107345.03
	First-tier	151796.49
	Second-tier	106040.57

13.4 分组的顺序

13.4.1 分组的顺序

之前我们在**分组/分组汇总**时，默认按分组字段升序排列。但是有的时候我们对结果集的**次序**不关心，并不需要对结果集**排序**。

按科目分组：

SID	Subject	Score
1	Math	93
2	Math	97
1	Chinese	79
2	Chinese	95

不排序

Subject	AvgScore
Math	95
Chinese	87

排序（按科目升序）

Subject	AvgScore
Chinese	87
Math	95

13.4.1 分组的顺序

【例13.18】查询2014年每个客户的订单中销售额前两名的记录。数据从文件Sales.txt中读取。

OrderID	Customer	OrderDate	Amount
10400	EASTC	2014/01/01	3063.0
10401	RATTC	2014/01/01	3868.6
10402	ERNSH	2014/01/02	2713.5
10403	ERNSH	2014/01/03	1005.9
10404	EASTC	2014/01/03	1675.0
10405	RATTC	2014/01/06	400.0
10406	EASTC	2014/01/07	2018.2
10407	ERNSH	2014/01/07	1194.0
…	…	…	…

13.4.1 分组的顺序

【例13.18】查询2014年每个客户的订单中销售额前两名的记录。数据从文件Sales.txt中读取。

解题思路

将销售表按客户分组，针对每个分组子集取销售额前两名的记录。函数 **A.group(xi, …)** 的**@u**选项用于在**分组后不需排序**的情况，不再针对分组字段**排序**。

保持原序

OrderID	Customer	OrderDate	Amount
10400	EASTC	2014/01/01	3063.0
10401	RATTC	2014/01/01	3868.6
10402	ERNSH	2014/01/02	2713.5
10403	ERNSH	2014/01/03	1005.9
10404	EASTC	2014/01/03	1675.0
10405	RATTC	2014/01/06	400.0
10406	EASTC	2014/01/07	2018.2
10407	ERNSH	2014/01/07	1194.0
…	…	…	…

Member
[[10400,EASTC,2014/01/01,3063.0], …]
[[10401,RATTC,2014/01/01,3868.6], …]
[[10402,EASTC,2014/01/02,2713.5], …]

OrderID	Customer	OrderDate	Amount
10400	EASTC	2014/01/01	3063.0
10404	EASTC	2014/01/03	1675.0
10406	EASTC	2014/01/07	2018.2

13.4.1 分组的顺序

【例13.18】查询2014年每个客户的订单中销售额前两名的记录。数据从文件Sales.txt中读取。

	A	B
1	=file("Sales.txt").import@t().select(year(OrderDate)==2014)	/从文件中导入销售表，并选出2014年的销售记录
2	=A1.group@u(Customer)	/使用group函数按客户分组，分组后不再排序
3	=A2.conj(~.top(-2; Amount))	/选出每个客户销售额前两名的订单

A2	Member
	[[10400,EASTC,2014/01/01,3063.0], …]
	[[10401,RATTC,2014/01/01,3868.6], …]
	…

A3	OrderID	Customer	OrderDate	Amount
	10400	EASTC	2014/01/01	3063.0
	10406	EASTC	2014/01/07	2018.2
	…	…	…	…

注意

- 本例中销售表是按时间有序的。加@u 选项时对结果集不排序。不加选项时默认会对结果集排序，也就是客户名称的顺序。

13.4.1 分组的顺序

【例13.19】计算每位员工每次值班的起止值班日期（例如 Emily 从 2018/03/01 至 2018/03/03算一次值班），值班表按日期升序。数据从文件 Duty.txt 中读取。

Date	Name
2018/03/01	Emily
2018/03/02	Emily
2018/03/03	Emily
2018/03/04	Johnson
2018/03/05	Ashley
2018/03/06	Emily
2018/03/07	Emily
2018/03/08	Ashley
2018/03/09	Ashley
2018/03/10	Ashley
…	…

13.4.1 分组的顺序

【例13.19】计算每位员工每次值班的起止值班日期（例如 Emily 从 2018/03/01 至 2018/03/03算一次值班），值班表按日期升序。数据从文件 Duty.txt 中读取。

解题思路

将连续出现的员工分在一组，当姓名字段发生改变时重新分组。函数 **A.group(xi, ...)** 的**@o** 选项只和相邻的对比，**结果集不再排序**。

姓名发生变化时重新分组

Date	Name
2018/03/01	Emily
2018/03/02	Emily
2018/03/03	Emily
2018/03/04	Johnson
2018/03/05	Emily
…	…

Date	Name
2018/03/01	Emily
2018/03/02	Emily
2018/03/03	Emily

Date	Name
2018/03/04	Johnson

Date	Name
2018/03/05	Emily

13.4.1 分组的顺序

【例13.19】计算每位员工每次值班的起止值班日期（例如 Emily 从 2018/03/01 至 2018/03/03算一次值班），值班表按日期升序。数据从文件 Duty.txt 中读取。

	A	B
1	=file("Duty.txt").import@t()	/从文件中导入值班表
2	=A1.group@o(Name)	/使用group函数的@o选项，当姓名改变时重新分组
3	=A2.new(Name,~(1).Date:Begin,~.m(-1).Date:End)	/因为日期有序，每组第一个日期作为起始时间，最后一个日期作为结束时间。

A2	Member
	[[2018/03/01,Emily], [2018/03/02,Emily], ...]
	[[2018/03/04, Johnson]]
	[[2018/03/05, Emily]]
	…

Date	Name
2018/03/01	Emily
2018/03/02	Emily
2018/03/03	Emily

A3	Name	Begin	End
	Emily	2018/03/01	2018/03/03
	Johnson	2018/03/04	2018/03/04
	Emily	2018/03/05	2018/03/05
	…	…	…

13.4.2 有序分组

对于有序集合来说，原始数据集的次序在分组时是否有意义呢？答案是肯定的。接下来我们分别学习三种有序分组：

- **序号分组**

  通过计算出来的序号分组。

- **值变化分组**

  分组字段值发生变化时重新分组。

- **条件变化分组**

  分组条件计算结果为true时重新分组。

13.4.2 有序分组

【例13.20】把日志文件整理成结构化数据（以USERID,UNAME,IP,TIME,URL,BROWSER,LOCATION,MODULE 为字段的序表）。数据从文件log.txt中读取。

现有日志的格式如下：

第一行是IP, TIME, GET, URL, BROWSER;

第二行是MODULE;

第三行是USERID, UNAME, LOCATION

```
10.10.10.143        2013-04-01 21:14:44        GET        /p/pt301/index.jsp  Mozilla/6.0
#module:production#
47356    Jessica    Chicago
10.10.2.76          2013-04-01 21:18:50        GET        /h/homepage.jsp     Chrome/35
#module:homepage#
419      Jacob      Houston
10.10.54.218        2013-04-01 21:25:19        GET        /p/pt27/index.jsp   Mozilla/6.0
#module:production#
3464     Madison    Detroit
10.10.10.145        2013-04-01 21:30:02        GET        /u/userlist/showlist.jsp          Mozilla/6.0
#module:usercenter#
432442   Phoenix    San Jose
10.2.1.242          2013-04-01 21:30:15        GET        /p/pt271/index.jsp  Mozilla/6.0
#module:production#
3435567  Megan      San Jose
```

13.4.2 有序分组

【例13.20】把日志文件整理成结构化数据（以USERID,UNAME,IP,TIME,URL,BROWSER,LOCATION,MODULE 为字段的序表）。数据从文件log.txt中读取。

解题思路

将日志格式每三行分为一组。然后将每组的成员按“\t”拆分，按位置生成结构化数据。函数 **A.group(x)** 当x 计算结果相同时分配到一组。

(序号-1)整除3，结果相同的分在同一组：

Index	Member
1	10.10.10.143 2013-04-01 21:14:44 GET /p/pt301/index.jsp Mozilla/6.0
2	#module:production#
3	47356 Jessica Chicago
4	10.10.2.76 2013-04-01 21:18:50 GET /h/homepage.jsp Chrome/35
5	#module:homepage#
6	419 Jacob Houston
…	…

结果为0

Index	Member
1	10.10.10.143 2013-04-01 21:14:44 GET /p/pt301/index.jsp Mozilla/6.0
2	#module:production#
3	47356 Jessica Chicago

结果为1

Index	Member
1	10.10.2.76 2013-04-01 21:18:50 GET /h/homepage.jsp Chrome/35
2	#module:homepage#
3	419 Jacob Houston

13.4.2 有序分组

【例13.20】把日志文件整理成结构化数据（以USERID,UNAME,IP,TIME,URL,BROWSER,LOCATION,MODULE 为字段的序表）。数据从文件log.txt中读取。

	A	B
1	=file("log.txt").read@n()	/从文件中按行读入字符串组成的序列
2	=A1.group((#-1)\3)	/使用group函数将每三行分为一组
3	=A2.(~.conj(~.split("\t")))	/每组按”\t”拆分每一行，合并为一个序列
4	=A3.new(~(7):USERID,~(8):UNAME,~(1):IP,~(2):TIME,~(4):URL,~(5):BROWSER,~(9):LOCATION,left(~(6).array("\:")(2),-1):MODULE)	/生成结构化数据

A2

Member
[[10.10.10.143 2013-04-01 21:14:44 GET /p/pt301/index.jsp Mozilla/6.0],…]
[[10.10.2.76 2013-04-01 21:18:50 GET /h/homepage.jsp Chrome/35],…]

Member
10.10.10.143 2013-04-01 21:14:44 GET …
#module:production#
47356 Jessica Chicago

A4

USERID	UNAME	IP	TIME	URL	BROWSER	LOCATION	MODULE
47356	Jessica	10.10.10.143	2013-04-01 21:14:44	/p/pt301/index.jsp	Mozilla/6.0	Chicago	production
419	Jacob	10.10.2.76	2013-04-01 21:18:50	/h/homepage.jsp	Chrome/35	Houston	homepage

【例13.21】查询蝉联奥运会奖牌榜第一名次数最多的是哪个国家，哪几届？数据从文件Olympic.txt中读取。

Game	Nation	Gold	Silver	Copper
30	USA	46	29	29
30	China	38	27	23
30	UK	29	17	19
30	Russia	24	26	32
30	Korea	13	8	7
30	Germany	11	19	14
30	France	11	11	12
30	Italy	8	9	11
30	Hungary	8	4	5
30	Australia	7	16	12
29	China	51	21	28
…	…	…	…	…

13.4.2 有序分组

【例13.21】查询蝉联奥运会奖牌榜第一名次数最多的是哪个国家，哪几届？数据从文件Olympic.txt中读取。（文件中记录了每届奥运会获得前十名国家的奖牌统计）

解题思路

数据按序号顺序进行**值变化分组**，当字段值发生变化时重新分组。

国家发生变化时重新分组

Game	Nation	Gold	Silver	Copper
30	USA	46	29	29
29	China	51	21	28
28	USA	35	39	29
27	USA	37	24	31
26	USA	44	32	25
…	…	…	…	…

Game	Nation	Gold	Silver	Copper
30	USA	46	29	29

Game	Nation	Gold	Silver	Copper
29	China	51	21	28

Game	Nation	Gold	Silver	Copper
28	USA	35	39	29
27	USA	37	24	31
26	USA	44	32	25

13.4.2 有序分组

值变化分组

【例13.21】查询蝉联奥运会奖牌榜第一名次数最多的是哪个国家，哪几届？数据从文件Olympic.txt中读取。（文件中记录了每届奥运会获得前十名国家的奖牌统计）

	A	B
1	=file("Olympic.txt").import@t()	/从文件中导入奥运会表
2	=A1.group@o1(Game).group@o(Nation)	/每届只保留第一名，使用group函数的@o选项，当第一名国家改变时重新分组
3	=A2.maxp(~.len())	/选出蝉联冠军次数最多的国家的记录

A2	Member
	[[30,USA,46,29,29]]
	[[29,China,51,21,28]]
	[[28,USA,35,39,29],[27,USA,37,24,31],...]
	[[25,Joint team (CIS),45,38,29]]
	…

Game	Nation	Gold	Silver	Copper
28	USA	35	39	29
27	USA	37	24	31
26	USA	44	32	25

A3	Game	Nation	Gold	Silver	Copper
	10	USA	41	32	30
	9	USA	22	18	16
	8	USA	45	27	27
	7	USA	41	27	28

13.4.2 有序分组

【例13.22】上证指数2019年最长连续上涨了多少天？数据从文件SSEC.txt中读取。（首个交易日认为指数上涨）

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
…	…	…	…

13.4.2 有序分组

【例13.22】上证指数2019年最长连续上涨了多少天？数据从文件SSEC.txt中读取。（首个交易日认为指数上涨）

解题思路

当条件（收盘价<=前日收盘价）发生改变时重新分组。函数 **A.group(x)** 的**@i** 选项，**x** 是bool表达式，如果 **x** 的结果为 true 则开始新的一组。选项**@i** 时只能有一个表达式 **x**。

收盘价不上涨时重新分组

Date	Open	Close	Amount
2019/01/02	2497.8805	2465.291	9.76E10
2019/01/03	2461.7829	2464.3628	1.07E11
2019/01/04	2446.0193	2514.8682	1.39E11
2019/01/07	2528.6987	2533.0887	1.46E11
2019/01/08	2530.3001	2526.4622	1.23E11
2019/01/09	2536.4173	2544.3449	1.61E11
2019/01/10	2543.8534	2535.0985	1.33E11
…	…	…	…

Date	Open	Close	Amount
2019/01/02	2497.8805	2465.291	9.76E10

Date	Open	Close	Amount
2019/01/03	2461.7829	2464.3628	1.07E11
2019/01/04	2446.0193	2514.8682	1.39E11
2019/01/07	2528.6987	2533.0887	1.46E11

Date	Open	Close	Amount
2019/01/08	2530.3001	2526.4622	1.23E11
2019/01/09	2536.4173	2544.3449	1.61E11

13.4.2 有序分组

【例13.22】上证指数2019年最长连续上涨了多少天？数据从文件SSEC.txt中读取。（首个交易日认为指数上涨）

	A	B
1	=file("SSEC.txt").import@t()	/从文件中导入上证指数表
2	=A1.select(year(Date)==2019).sort(Date)	/选出2019年数据并按日期排序
3	=A2.group@i(Close<=Close[-1])	/使用group函数的@i选项，当条件变化（当日收盘价没有上涨）时重新分组。这样分在一组的都是连续上涨的记录。
4	=A3.max(~.len())	/计算最大天数

A3	Member
	[[2019/01/02,2497.8805,2465.291,9.76E10]]
	[[2019/01/03,2461.7829,2464.3628,1.07E11], …]
	[[2019/01/08,2530.3001,2526.4622,1.23E11], ...]
	…

Date	Open	Close	Amount
2019/01/03	2461.7829	2464.3628	1.07E11
2019/01/04	2446.0193	2514.8682	1.39E11
2019/01/07	2528.6987	2533.0887	1.46E11

A4	Value
	7

13.5 习题

13.5 习题

【习题13.1】统计各班每个学生的总分。数据从文件Score.txt 中读取。

CLASS	STUDENTID	SUBJECT	SCORE
1	1	English	84
1	1	Math	77
1	1	Chinese	69
2	1	English	96
2	1	Math	63
2	1	Chinese	81
1	2	English	81
1	2	Math	80
1	2	Chinese	97
…	…	…	…

13.5 习题

【习题13.1】统计各班每个学生的总分。数据从文件Score.txt 中读取。

解题思路

可以使用 **A.groups()** 函数按班级和学生编号两个字段**分组汇总**，统计分数之和。

	A	B
1	=file("Score.txt").import@t()	/从文件中导入成绩表
2	=A1.groups(CLASS, STUDENTID; sum(SCORE):TotalScore)	/分组汇总计算各班每个学生的总分

A2	CLASS	STUDENTID	TotalScore
	1	1	230
	1	2	258
	1	3	228
	1	4	240
	…	…	…

13.5 习题

【习题13.2】统计2014年每种产品的销售次数。数据从文件Sales.txt 中读取。

OrderID	ProductID	OrderDate	Amount
10400	27	2014/01/01	3063.0
10401	17	2014/01/01	3868.6
10402	70	2014/01/02	2713.5
…	…	…	…

13.5 习题

【习题13.2】统计2014年每种产品的销售次数。数据从文件Sales.txt 中读取。

解题思路

可以使用 A.groups() 函数按产品字段**分组汇总**，统计数量。

	A	B
1	=file("Sales.txt").import@t().select(year(OrderDate)==2014)	/从文件中导入销售表，并选出2014年记录
2	=A1.groups(ProductID; count(~):Count)	/分组汇总每个产品的销售次数

A2	ProductID	Count
	1	7
	2	4
	3	7
	…	…

13.5 习题

【习题13.3】2014年有销售记录的客户有哪些。数据从文件Sales.txt 中读取。

OrderID	Customer	OrderDate	Amount
10400	EASTC	2014/01/01	3063.0
10401	HANAR	2014/01/01	3868.6
10402	ERNSH	2014/01/02	2713.5
…	…	…	…

13.5 习题

【习题13.3】2014年有销售记录的客户有哪些。数据从文件Sales.txt 中读取。

解题思路

可以使用函数 **A.id()** 去重。

	A	B
1	=file("Sales.txt").import@t().select(year(OrderDate)==2014)	/从文件中导入销售表，并选出2014年记录
2	=A1.id(Customer)	/使用id 函数按客户名称去重，返回客户名称的唯一值

A2	Member
	ANATR
	ANTON
	AROUT
	…

13.5 习题

【习题13.4】查看总分高于240分的学生各科成绩。数据从文件Score.txt 中读取。

CLASS	STUDENTID	SUBJECT	SCORE
1	1	English	84
1	1	Math	77
1	1	Chinese	69
…	…	…	…

13.5 习题

【习题13.4】查看总分高于240分的学生各科成绩。数据从文件Score.txt 中读取。

解题思路

可以使用 **A.group()** 函数按学生编号**分组**，选出满足条件（总分高于240）的**分组子集。**

	A	B
1	=file("Score.txt").import@t()	/从文件中导入成绩表
2	=A1.group(STUDENTID)	/按学生编号分组
3	=A2.select(~.sum(SCORE)>240).conj()	/选出总分高于240的分组子集，然后合并

A2

Member
[[1,1,English,84],[1,1,Math,77],[1,1, Chinese,69]]
[[1,2,English,97],[1,2,Math,80],[1,2,Chinese,81]]
[[1,3,English,75],[1,3,Math,86],[1,3,Chinese,67]]
…

A3

CLASS	STUDENTID	SUBJECT	SCORE
1	2	English	97
1	2	Math	80
1	2	Chinese	81
…	…	…	…

13.5 习题

【习题13.5】计算出每个用户最后登录前三天内的登录次数。数据从文件LoginLog.txt 中读取。

ID	UserID	IP	LoginTime
1	1	37.17.184.11	2012/05/09 09:01:10
2	3	61.134.201.1	2012/05/09 09:02:43
3	7	124.114.171.101	2012/05/09 09:03:18
…	…	…	…

13.5 习题

【习题13.5】计算出每个用户最后登录前三天内的登录次数。数据从文件LoginLog.txt 中读取。

解题思路

可以使用 **A.group()** 函数按用户ID**分组**，在每个**分组子集**中找到最后登录时间，再计算该时间前三天内的登录次数。

	A	B
1	=file("LoginLog.txt").import@t()	/从文件中导入登录日志表
2	=A1.group(UserID)	/按用户编号分组
3	=A2.new(UserID, ~.max(LoginTime):Last, ~.count(interval(LoginTime, Last)<=3):Count)	/计算每个用户的最后登录时间，并统计登录时间与最后登录时间相差3天以内的记录的数量。

A2	Member
	[[1,1,37.17.184.11,2012/05/09 09:01:10], ...]
	[[4,2,183.202.48.25,2012/05/09 09:05:15], ...]
	…

A3	UserID	Last	Count
	1	2012/05/09 09:12:41	2
	2	2012/05/09 09:09:41	2
	…	…	…

ID	UserID	IP	LoginTime
4	2	183.202.48.25	2012/05/09 09:05:15
…	…	…	…

13.5 习题

【习题13.6】查看英语成绩最高的三个学生的所有成绩。数据从文件Score.txt中读取。

CLASS	STUDENTID	SUBJECT	SCORE
1	1	English	84
1	1	Math	77
1	1	Chinese	69
…	…	…	…

13.5 习题

【习题13.6】查看英语成绩最高的三个学生的所有成绩。数据从文件Score.txt中读取。

解题思路

首先查询英语成绩最高的三个学生，再按此顺序进行**对齐分组**。可以使用 **P.align(A,y)** 函数的**@a** 选项，返回所有匹配的成员。

	A	B
1	=file("Score.txt").import@t()	/从文件中导入成绩表
2	=A1.select(SUBJECT:"English").top(-3; SCORE)	/选出英语成绩前三名的记录
3	=A2.(STUDENTID)	/选出前三名的学生ID
4	=A1.align@a(A3, STUDENTID).conj()	/成绩表按前三名的学生ID对齐分组

A3	Member
	4
	9
	6

A4	CLASS	STUDENTID	SUBJECT	SCORE
	1	4	English	96
	1	4	Math	63
	1	4	Chinese	81
	…	…	…	…

13.5 习题

【习题13.7】按学生各科平均分查询“优良中差”各有多少人。数据从文件Score.txt中读取。

CLASS	STUDENTID	SUBJECT	SCORE
1	1	English	84
1	1	Math	77
1	1	Chinese	69
…	…	…	…

85—100为优秀，75—84为良好， 60—74为中等， 60以下为差

13.5 习题

【习题13.7】按学生各科平均分查询“优良中差”各有多少人。数据从文件Score.txt中读取。

解题思路

首先**分组汇总**，统计每位学生的各科平均分，再按照条件进行**枚举分组**。使用函数P.enum(E,y) 的@n 选项，会在结果集最后创建一个新组，用于存放未匹配的成员。

	A	B
1	=file("Score.txt").import@t()	/从文件中导入成绩表
2	[?>=85,?>=75,?>=60]	/创建分组条件的常数序列
3	=A1.groups(STUDENTID; avg(SCORE):AvgScore)	/分组汇总，统计每位学生的平均分
4	=A3.enum@n(A2,AvgScore)	/学生平均分按A2的条件枚举分组
5	=A4.new(["Excellent","Good","Normal","Fail"](#):Level, ~.len():Count)	/统计优良中差的人数

A3	STUDENTID	AvgScore
	1	76.67
	2	86.0
	3	76.0
	…	…

A4	Member
	[2]
	[1,3,4,6,7,9,10]
	[5,8,11,12,13,14]
	[]

A5	Level	Count
	Excellent	1
	Good	7
	Normal	6
	Fail	0

13.5 习题

【习题13.8】在一个电影院中，想统计相邻的空座最多有多少。数据从文件Cinema.txt 中读取。

Row	Col	Empty
1	1	YES
1	2	YES
1	3	NO
…	…	…

13.5 习题

【习题13.8】在一个电影院中，想统计相邻的空座最多有多少。数据从文件Cinema.txt 中读取。

解题思路

这是**有序分组**的问题。可以使用 **A.group()** 函数的**@o** 选项只和相邻的对比， 当行数发生变化（前后排的不算相邻），或者有座无座发生变化时，重新分组。

	A	B
1	=file("Cinema.txt").import@t()	/从文件中导入电影院订位表
2	=A1.group@o(Row,Empty; ~.count():Count)	/使用group函数的@o 选项，值变化时重新分组
3	=A2.select(Empty:"YES").max(Count)	/选出空座的分组子集，计算最多的成员数

A2	Row	Empty	Count
	1	YES	2
	1	NO	4
	1	YES	3
	…	…	…

A3	Value
	7

13.5 习题

【习题13.9】求上证指数2019年连续放量上涨的最长天数。数据从文件SSEC.txt中读取。（首个交易日认为放量上涨）

Date	Open	Close	Amount
2019/12/31	3036.3858	3050.124	2.27E11
2019/12/30	2998.1689	3040.0239	2.67E11
2019/12/27	3006.8517	3005.0355	2.58E11
2019/12/26	2981.2485	3007.3546	1.96E11
…	…	…	…

13.5 习题

【习题13.9】求上证指数2019年连续放量上涨的最长天数。数据从文件SSEC.txt中读取。（首个交易日认为放量上涨）

解题思路

这是**有序分组**的问题。可以使用 **A.group()** 函数的**@i** 选项，当条件变化时（收盘价没有上涨或者交易量没有上涨）重新分组。

	A	B
1	=file("SSEC.txt").import@t()	/从文件中导入上证指数表
2	=A1.select(year(Date)==2019).sort(Date)	/选出2019年的记录，并按日期排序
3	=A2.group@i(Close<=Close[-1] || Amount<=Amount[-1])	/使用group函数的@i 选项，条件变化分组
4	=A3.max(~.len())	/计算最大的成员数，即最长天数

A3	Member
	[[2019/01/02,2497.8805,2465.291,9.76E10]]
	[[2019/01/03,2461.7829,2464.3628,1.07E11], …]
	…

Date	Open	Close	Amount
2019/01/03	2461.7829	2464.3628	1.07E11
…	…	…	…

A4	Value
	4

13.5 习题

【习题13.10】求2019年各类指数连续上涨的最长天数。数据从文件StockIndex.txt中读取。（首个交易日认为指数上涨）

Date	Code	Open	Close	Amount
2019/1/2	399001	7259.489	7149.2665	63756033352
2019/1/2	399006	1250.623	1228.7668	14891216036
2019/1/2	399007	3046.3877	2998.0988	51134636181
…	…	…	…	…

13.5 习题

【习题13.10】求2019年各类指数连续上涨的最长天数。数据从文件StockIndex.txt中读取。（首个交易日认为指数上涨）

解题思路

首先按指数代码进行**分组**，接下来对每个**分组子集**进行**有序分组**。使用 **A.group()** 函数的**@u** 选项分组，不对结果集进行排序。使用 **A.group()** 函数的**@i** 选项，当条件变化时（收盘价没有上涨）重新分组。

	A	B
1	=file("StockIndex.txt").import@t().select(year(Date)==2019)	/从文件中导入指数表，并选出2019年记录
2	=A1.group@u(Code)	/按指数代码分组，不排序
3	=A2.(~.sort(Date).group@i(Close<=Close[-1]))	/对每个分组子集使用group函数的@i 选项，条件变化分组
4	=A3.new(~.Code:Code, ~.max(~.len()):Day)	/分别计算每个指数连续上涨的最大天数

13.5 习题

【习题13.10】求2019年各类指数连续上涨的最长天数。数据从文件StockIndex.txt中读取。(首个交易日认为指数上涨)

A2	Member
	[[2019/01/02,399001,7259.489,7149.2665,6.38E10], …]
	[[2019/01/02,399006,1250.623,1228.7668,1.49E10], …]
	…

Date	Code	Open	Close	Amount
2019/01/02	399001	7259.489	7149.2665	6.38E10
2019/01/03	399001	7134.1461	7089.4406	6.99E10
…	…	…	…	…

A3	Member
	[[[2019/01/02,399001,7259.489,7149.2665,6.38E10], …],…]
	[[[2019/01/02,399006,1250.623,1228.7668,1.49E10], …],…]
	…

Member
[[2019/01/02,399001,7259.489,7149.2665,6.38E10]]
[[2019/01/03,399001,7134.1461,7089.4406,6.99E10], …]
…

值得注意的是，两次分组后，A3的值是集合的集合的集合。

Date	Code	Open	Close	Amount
2019/01/03	399001	7134.1461	7089.4406	6.99E10
2019/01/04	399001	7022.1229	7284.8403	9.38E10
2019/01/07	399001	7333.3171	7400.2046	9.95E10

A4	Code	Day
	399001	7
	399006	7
	…	…

第14章 表间关联

14.1 主键与索引

- 根据主键查找
- 索引表

14.2 表间关联

- 表间关联
- 常见的关联关系

14.3 外键表

14.4 同维表和主子表

14.5 交叉连接与非等值连接

- 交叉分组
- 非等值连接

14.6 习题

14.1 主键与索引

14.1.1 根据主键查找

主键是序表中的一个或多个字段，它的值用于**唯一地标识**序表中的某一条记录。

ID是员工表的主键，每一个ID都能标识**唯一**一名员工。查找ID是2的员工：

ID	Name	Birthday	EntryDate	Department	Salary
1	Rebecca	1974/11/20	2005/03/11	6	7000
2	Ashley	1980/07/19	2008/03/16	2	11000
3	Rachel	1970/12/17	2010/12/01	7	9000

14.1.1 根据主键查找

【例14.1】查询订单编号为28571610的订单销售额。数据从文件Sales.txt中读取：

OrderID	Customer	SellerId	OrderDate	Amount
81182311	VINET	5	2013/07/04	2440.0
98807954	TOMSP	6	2013/07/05	1863.4
…	…	…	…	…
28571610	MAISD	3	2015/04/07	295.38
…	…	…	…	…

14.1.1 根据主键查找

【例14.1】查询订单编号为28571610的订单销售额。数据从文件Sales.txt中读取：

OrderID	Customer	SellerId	OrderDate	Amount
81182311	VINET	5	2013/07/04	2440.0
98807954	TOMSP	6	2013/07/05	1863.4
…	…	…	…	…
28571610	MAISD	3	2015/04/07	295.38
…	…	…	…	…

解题思路

可以利用表的**主键值**查找到指定的记录。

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=A1.keys(OrderID)	/设置销售表的主键为OrderID
3	=A2.find(28571610)	/使用函数A.find(k)，查找主键为28571610的销售记录
4	=A3.Amount	/从销售记录中取出销售额

A4
295.38

注意

- 函数 **A.find(k)** 用于根据主键值k查找匹配的记录。

14.1.1 根据主键查找

【例14.1】查询订单编号为28571610的订单销售额。数据从文件Sales.txt中读取：

Index	OrderID	Customer	SellerId	OrderDate	Amount
1	81182311	VINET	5	2013/07/04	2440.0
2	98807954	TOMSP	6	2013/07/05	1863.4
…	…	…	…	…	…
757	28571610	MAISD	3	2015/04/07	295.38
…	…	…	…	…	…

解题思路

还可以先查找表的主键值所在记录的序号，再进行其他计算。

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=A1.keys(OrderID)	/设置销售表的主键为ID
3	=A2.pfind(28571610)	/使用函数A.pfind(k)，查找主键为28571610的销售记录所在行号
4	=A2(A3).Amount	/从销售记录中取出销售额

A3
757

A4
295.38

注意

- 函数 **A.pfind(k)** 用于根据主键值k查找匹配的记录序号。

14.1.1 根据主键查找

根据主键查找缺省是遍历查找的，需要逐条比较，直到发现目标记录为止。

顺序比较主键值是否等于28571610。第757行时匹配主键值，共比较757次。

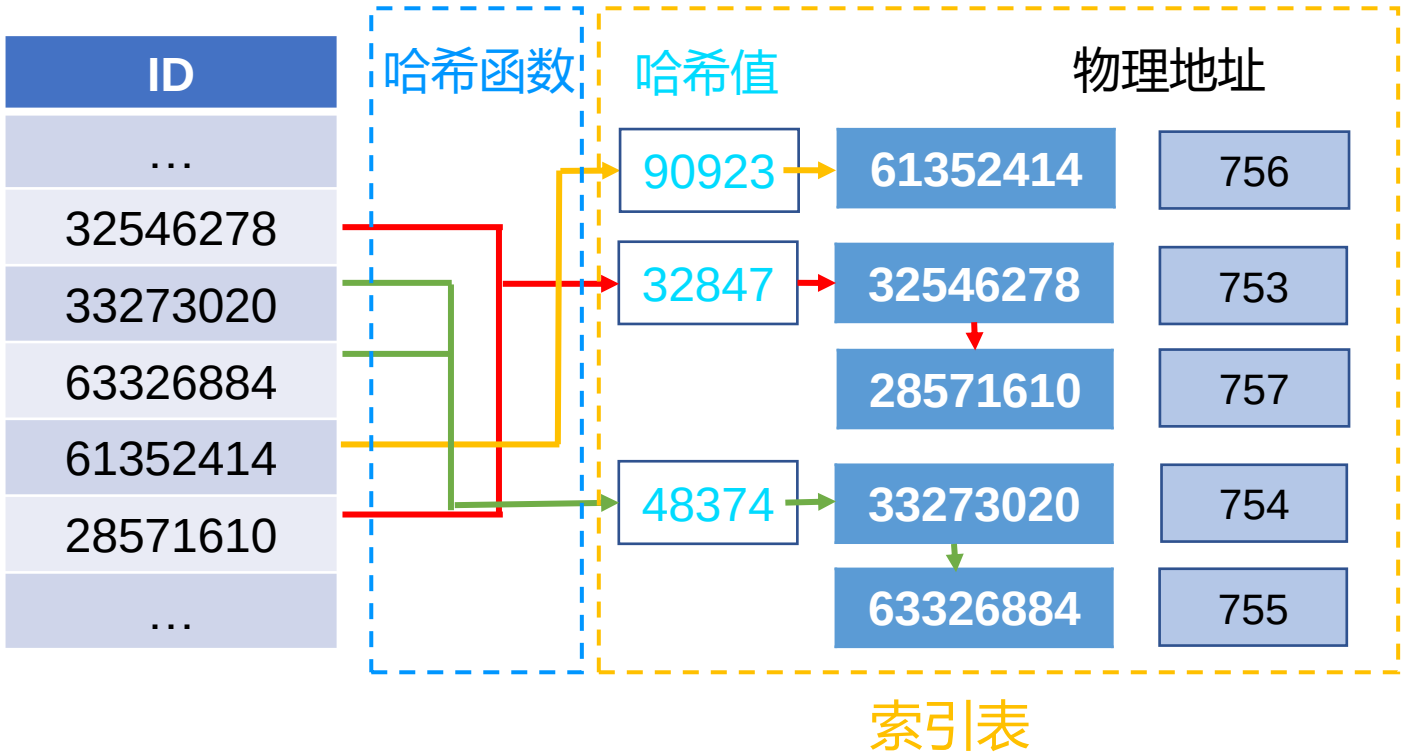
OrderID	Customer	SellerId	OrderDate	Amount
81182311	VINET	5	2013/07/04	2440.0
98807954	TOMSP	6	2013/07/05	1863.4
…	…	…	…	…
28571610	MAISD	3	2015/04/07	295.38
…	…	…	…	…

◀757行

遍历查找的效率很差。对于N条记录的序表，平均要比较N/2次。

14.1.2 索引表

为了解决遍历查找效率差的问题，我们可以使用索引表。索引的作用相当于图书的目录，可以根据目录中的页码快速找到所需的内容。



14.1.2 索引表

根据**主键**生成**索引表**，按**主键**查找时，会明显的提高效率。

- 遍历查找：

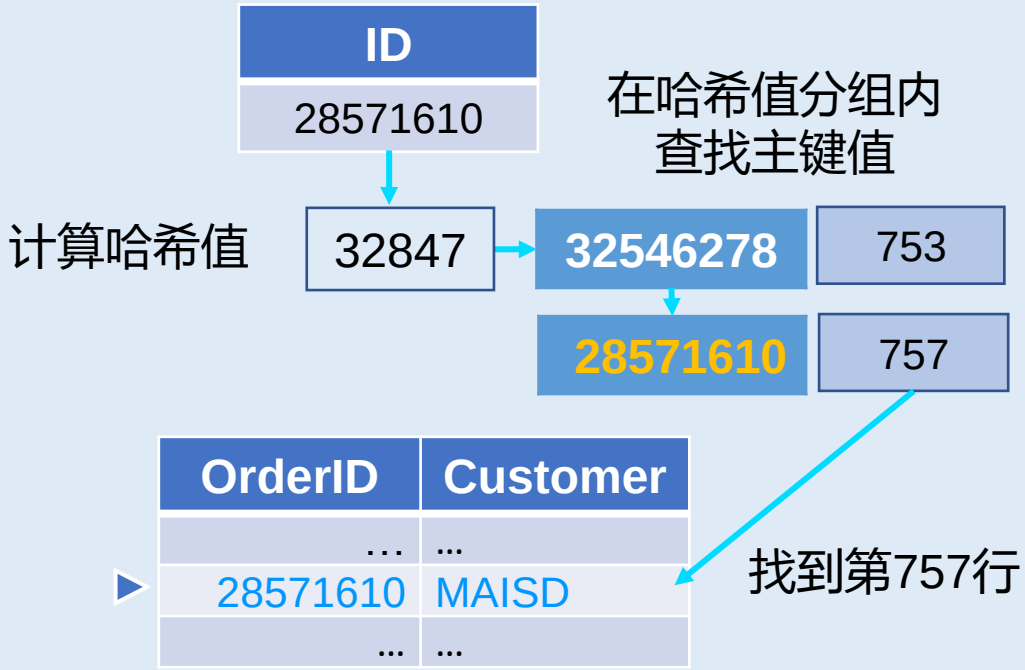
N条记录的序表，平均查找N/2次。数据量大时查找效率明显降低。

遍历查找757次

Index	OrderID	Customer
1	81182311	VINET
2	98807954	TOMSP
…	…	…
757	28571610	MAISD
…	…	…

- 索引查找：

按哈希值分为k组，平均查找N/2k次，当k越大时查找效率越高。数据量大时明显优于遍历查找。



14.1.2 索引表

【例14.1】查询订单编号为28571610的订单销售额。如何提高查找效率?

OrderID	Customer	SellerId	OrderDate	Amount
81182311	EASTC	1	2014/01/01	3063.0
98807954	HANAR	1	2014/01/01	3868.6
…	…	…	…	…
28571610	BLAUS	4	2015/07/17	3774.05
…	…	…	…	…

解题思路

创建**主键索引**，在根据主键查找时可以提高效率，快速定位。

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=A1.keys@i(OrderID)	/设置销售表主键为ID并建索引
3	=A2.find(28571610)	/使用函数A.find(k)，查找主键为28571610的销售记录
4	=A3.Amount	/从销售记录中取出销售额

A4
3774.05

注意

- 函数**T.keys@i(k)** 用于设置主键时创建主键索引。索引表只需要创建一次，有关主键的处理会自动使用索引。

14.2 表间关联

14.2.1 表间关联

员工表中的 Department 字段存储着员工所属部门的编码，这些编码或者为空，或者来自于部门表ID字段的值。

员工表

ID	Name	Department
1	Rebecca	6

部门表

ID	Name
6	R&D

员工表和部门表之间存在的数据相互依赖关系，就叫着表间关联关系。**表间关联**关系可以分为以下三种：

- 一对一
- 一对多
- 多对多

14.2.1 表间关联

一对一关联：一张表的一条记录只能与另外一张表的一条记录进行对应；反之亦然。

学生表

ID	Name	Birthday
1	Rebecca	2010/09/08

1 : 1

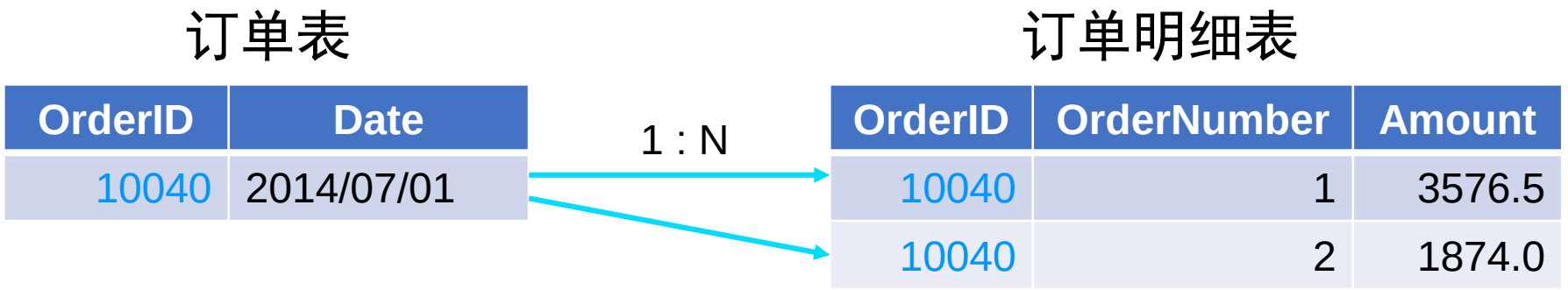
学生联系表

ID	Contacts	Address
1	Mrs. Moore	124 Guangming North Road

如上图，学生表中每个ID都是唯一的，学生联系表中每个ID也是唯一的，这样学生表中的一条记录只能和学生联系表的一条记录对应，反过来也是这样，所以学生表和学生联系表是一对一的关联关系。

14.2.1 表间关联

一对多关联：一张表的一条记录能与另外一张表的任意条记录进行对应。



上图中，订单表和订单明细表就是一对多的关系，每个订单可能有任意条订单明细。

14.2.1 表间关联

多对多关联：表 A 的一条记录能够对应表 B 中的任意条记录；同时表 B 中的一条记录也能够对应表 A 中的任意条记录。

矩阵A

row	col	value
1	1	1
1	2	2
2	1	3
2	2	4

M : N

row	col	value
1	1	1
1	2	3
2	1	2
2	2	4

矩阵B

以矩阵乘法为例，记C=AB，则$C_{ij} = \sum_{k=0}^{n}(A_{ik} * B_{kj})$，即$C_{ij}$是通过矩阵A的第i行各列与矩阵B的第j列各行对位相乘然后求和得来的，因此求解时需要找出矩阵A的列与矩阵B的行的所有对应，如上图所示，每一种颜色的线条表示一组A的列与B的行的对应。

详细实现过程参见第72页【例14.14】。

14.2.2 常见的关联关系

存在**关联关系**的表与表之间可以通过**关联字段**进行连接。通过关联字段连接时，过滤条件是一个或多个相等关系则称为**等值连接**。

员工表

ID	Name	Department
1	Rebecca	6

部门表

ID	Name
6	R&D

例如，连接员工表和部门表时，以 Employee.Department == Department.ID 作为连接条件，就是一个等值连接。

等值连接是最常用的连接形式，常用于以下几种关联关系：

- 外键表
- 同维表
- 主子表

14.2.2 常见的关联关系

表 A 的某些字段与表 B 的**主键**关联。A 表中与 B 表主键关联的字段称为指向B表的**外键**，B 也称为 A 的**外键表**。表 A 与其外键表 B 是**多对一**的关系。

员工表

EMP_ID	EMP_NAME	DEPT_ID
1	Rebecca	2
2	Ashley	2
3	Rachel	1

部门表

DEPT_ID	DEPT_NAME
1	Admin
2	R&D

员工表中的部门字段称为指向部门表的**外键**。

14.2.2 常见的关联关系

事实表包含描述业务的数据，还包含了指向维表（维度表）的外键。维表包含了事实表中指定属性的相关详细信息。

日常考勤表

ID	Time	EMP_ID
1	09:05:32	2
2	09:15:17	1
3	18:01:55	2

员工表

ID	EMP_NAME	DEPT_ID
1	Rebecca	2
2	Ashley	2
3	Rachel	1

部门表

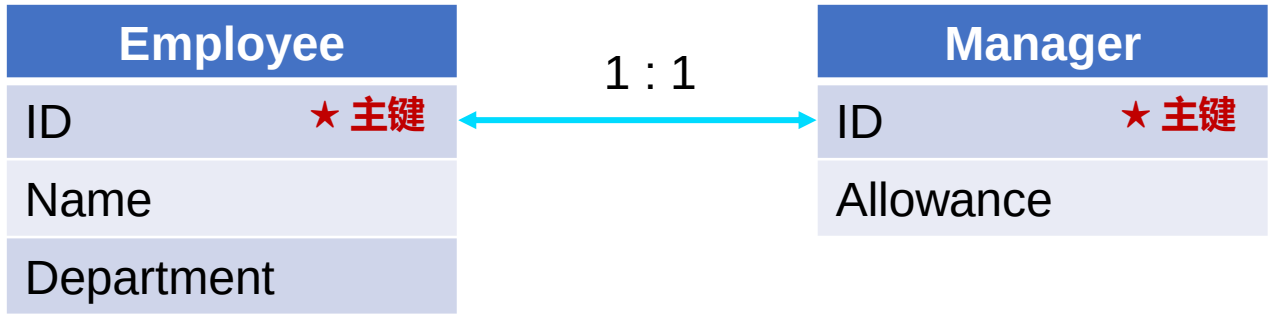
ID	DEPT_NAME
1	Admin
2	R&D

员工表中记录了员工的基本信息，是事实表；部门表中记录了员工部门的描述性信息，是维表。

事实表和维表的关系是相对的。对于日常考勤表和员工表来说，日常考勤表是事实表，而员工表则变成了维表。

14.2.2 常见的关联关系

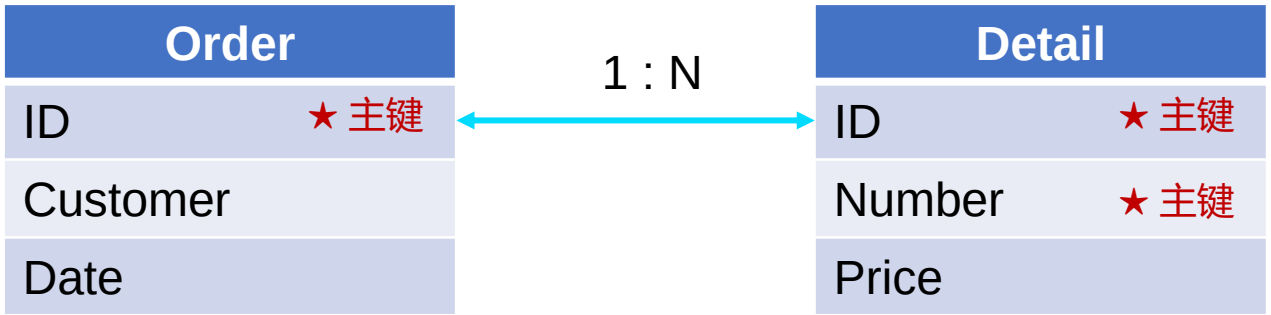
表 A 的主键与表 B 的主键关联，A 和 B 互称为同维表。同维表之间是一对一的关系，同维表之间的关系是对等的。



上图中，员工表和经理表是同维表。经理也是员工，只是比员工多了一些属性。

14.2.2 常见的关联关系

表 B 主键的部分字段与表 A 的主键关联，A 称为主表，B 称为子表。主表和子表之间是一对多的关系。主子表关系是不对等的，从子表引用主表则和外键表类似。



上图中，订单表和订单明细表是主子表。

14.2.2 常见的关联关系

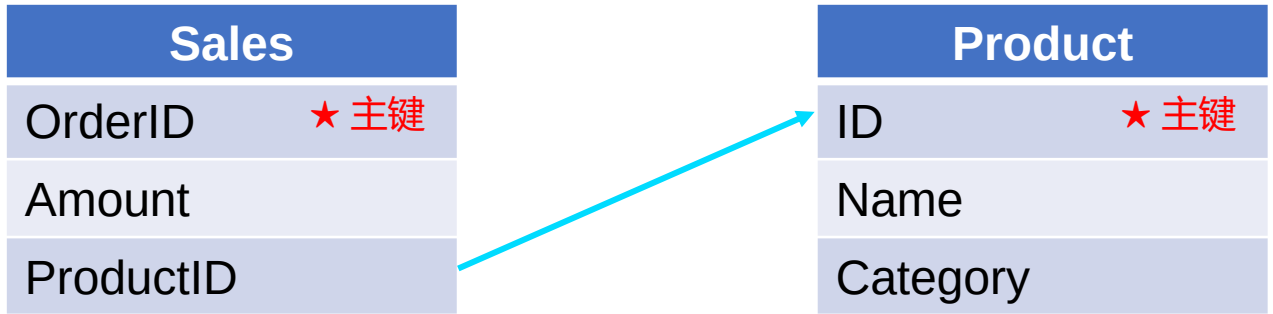
外键表、同维表、主子表都和**主键**相关，并分别对应了多对一、一对一或一对多的关系。**无主键**相关时，会导致出现**多对多**的关系，在实际应用场景中多对多关系一般是业务错误。

注意

- 本节中说的**主键**是指**逻辑上的主键**，也就是在表中**取值唯一**的字段（组）。不一定是物理表上建立的主键。

14.3 外键

【例14.2】查询销售金额超过1000的销售单详情。销售表和产品表数据分别从文件Sales.txt 和Products.txt 中读取：



【例14.2】查询销售金额超过1000的销售单详情。销售表和产品表数据分别从文件Sales.txt 和Products.txt 中读取：

解题思路

通过外键字段与外键表的主键进行连接，增加需要的外键表字段。

销售表

OrderID	Amount	ProductID
81182311	2440.0	59

产品表中查找到ID为5的记录

ID	Name	Category
59	Bright cheese	4

OrderID	Amount	ProductID	ProductName
81182311	2440.0	59	Bright cheese

增加外键表字段：产品名称

【例14.2】查询销售金额超过1000的销售单详情。销售表和产品表数据分别从文件Sales.txt 和Products.txt 中读取：

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=file("Products.txt").import@t().keys@i(ID)	/从文件中导入产品表
3	=A1.select(Amount>1000)	/选出销售额大于1000的记录
4	=A3.derive(A2.find(ProductID).Name:ProductName)	/根据产品ID在产品表中查找对应记录，并将产品名称字段增加到销售表上。

A4	OrderID	Amount	ProductID	ProductName
	81182311	2440.0	59	Bright cheese
	98807954	1863.4	38	Green tea
	…	…	…	…

注意

- 函数**A.derive(xi:Fi,…)** 用于生成新序表，先抄录原有字段。

14.3 外键

【例14.2】查询销售金额超过1000的销售单详情。销售表和产品表数据分别从文件Sales.txt和Products.txt 中读取：

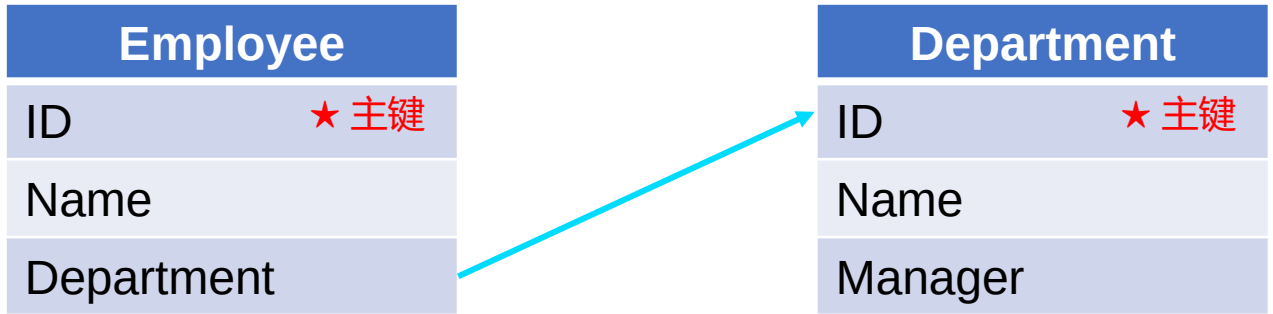
	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=file("Products.txt").import@t()	/从文件中导入产品表
3	=A1.select(Amount>1000)	/选出销售额大于1000的记录
4	=A3.join(ProductID, A2:ID, Name:ProductName)	/我们可以使用A.join()函数来简化，与外键表连接后增加外键表字段。

A4	OrderID	Amount	ProductID	ProductName
	81182311	2440.0	59	Bright cheese
	98807954	1863.4	38	Green tea
	…	…	…	…

注意

- 函数A.join(C,T:K,x:F,…; …)通过序表A的字段C连接序表T的字段K，增加表T的x字段（当x为~时即是增加外键引用记录）到表A的F字段。

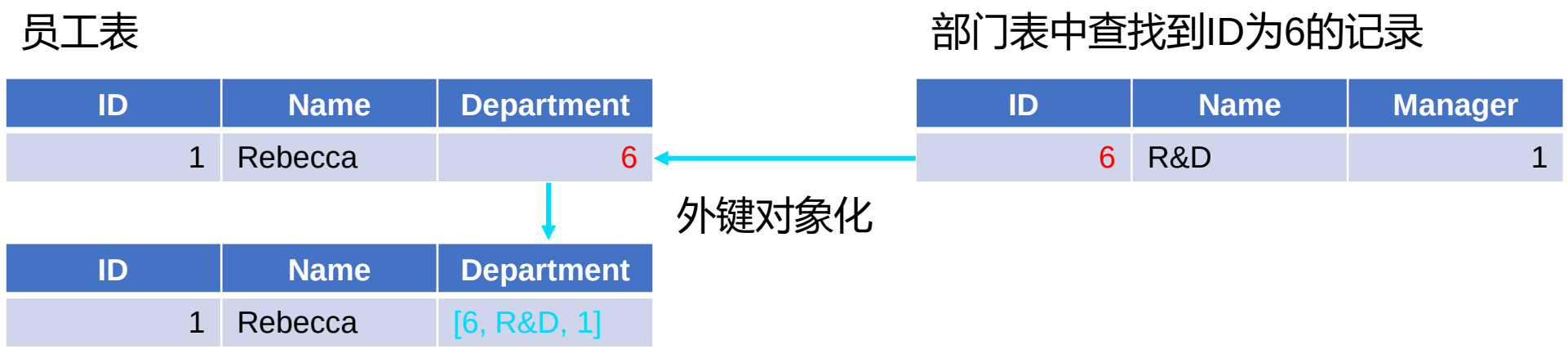
【例14.3】查询员工姓名及其所在部门名称。员工表和部门表数据分别从文件Employee.txt 和Department.txt 中读取：



【例14.3】查询员工姓名及其所在部门名称。员工表和部门表数据分别从文件Employee.txt 和Department.txt 中读取：

解题思路

可以通过**外键对象化**直接引用。根据外键字段值，在外键表中查找相应的记录，并设置到外键字段上（会改变原值）。这样就可以当作单表处理。



14.3 外键

【例14.3】查询员工姓名及其所在部门名称。员工表和部门表数据分别从文件Employee.txt和Department.txt 中读取：

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=file("Department.txt").import@t().keys@i(ID)	/从文件中导入部门表，设置主键为ID并建立索引
3	=A1.run(Department=A2.find(Department))	/循环员工表，在部门表中查找相应的记录引用并设置到部门字段
4	=A3.new(Name, Department.Name:DeptName)	/创建序表，返回员工姓名和部门名称

A3

ID	Name	Department
1	Rebecca	[6, R&D, 1]
…	…	…

ID	Name	Manager
6	R&D	1

A4

Name	DeptName
Rebecca	R&D
Ashley	Finance
…	…

注意

- 经过外键对象化，外键字段值转换成外键表中对应记录。对应的记录被修改时，会同步影响这两个表。

14.3 外键

【例14.3】查询员工姓名及其所在部门名称。员工表和部门表数据分别从文件Employee.txt 和Department.txt 中读取：

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=file("Department.txt").import@t().keys@i(ID)	/从文件中导入部门表，设置主键为ID并建立索引
3	=A1.switch(Department, A2:ID)	/我们还可以通过switch函数来简化，实现外键对象化。
4	=A3.new(Name, Department.Name:DeptName)	/创建序表，返回员工姓名和部门名称

A3

ID	Name	Department
1	Rebecca	[6, R&D, 1]
…	…	…

ID	Name	Manager
6	R&D	1

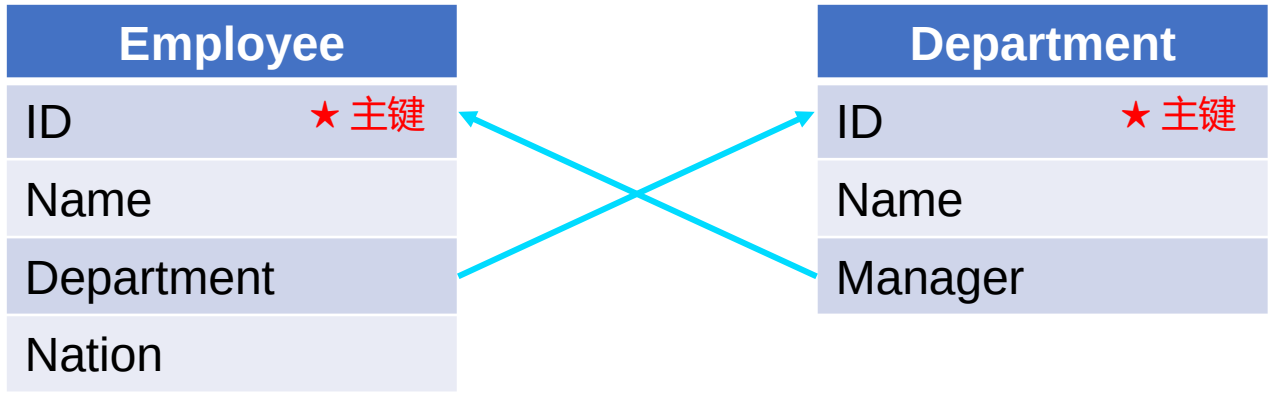
A4

Name	DeptName
Rebecca	R&D
Ashley	Finance
…	…

注意

- 函数A.switch(F, T:x) 用于将序列A的F字段与序列T的x字段连接（x省略时使用T的主键连接），转换成T中对应的记录。

【例14.4】查询哪些美国籍员工有一个中国籍经理。员工表和部门表数据分别从文件Employee.txt 和Department.txt 中读取：



【例14.4】查询哪些美国籍员工有一个中国籍经理。员工表和部门表数据分别从文件Employee.txt 和Department.txt 中读取：

解题思路

对于多层外键表连接，可以逐层**外键对象化**，最终可以当作单表处理。

部门表

ID	Name	Manager
4	Marketing	6

外键对象化

ID	Name	Manager
4	Marketing	[6, Matthew, 4]

员工表

ID	Name	Department
8	Megan	4

外键对象化

ID	Name	Department
8	Megan	[4, Marketing, […]]

【例14.4】查询哪些美国籍员工有一个中国籍经理。员工表和部门表数据分别从文件Employee.txt 和Department.txt 中读取：

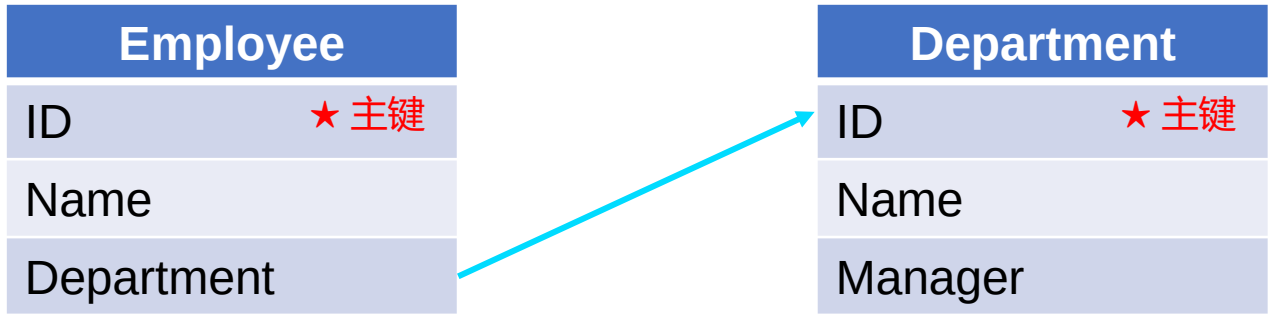
	A	B
1	=file("Employee.txt").import@t().keys(ID)	/从文件中导入员工表
2	=file("Department.txt").import@t().keys(ID)	/从文件中导入部门表
3	=A2.switch(Manager, A1)	/将部门表的经理字段转换为员工表中对应的员工记录。
4	=A1.switch(Department, A2)	/将员工表的部门字段转换为部门表中对应的部门记录。
5	=A4.select(Nation=="American" && Department.Manager.Nation=="Chinese")	/选出经理是中国籍的美国籍员工。

A5	ID	Name	Department
	8	Megan	4
	27	Alexis	4
	…	…	…

注意

- 本例中员工表和部门表进行了循环引用，员工表中的部门字段值转换成部门表中对应记录，部门表中的经理字段值转换成员工表中对应记录。

【例14.5】查询所有员工的姓名和部门名称，员工的部门不在部门表中的显示原值。员工表和部门表数据分别从文件Employee.txt 和Department.txt 中读取：



14.3 外键

【例14.5】查询所有员工的姓名和部门名称，员工的部门不在部门表中的显示原值。员工表和部门表数据分别从文件Employee.txt 和Department.txt 中读取：

解题思路

当所属部门的指向为空时，外键对象化会使该字段的代码值丢失。
解决方法1：不使用外键对象化，增加需要的外键表字段。
解决方法2：仍然外键对象化，但是外键对象产生在新增字段上。

外键指向相应记录

ID	Name	Department
1	Rebecca	6

新增外键表字段

ID	Name	Department	DEPT_NAME
1	Rebecca	6	R&D

ID	Name	Department
1	Rebecca	6

新增外键对象

ID	Name	Department	DEPT
1	Rebecca	6	[6, R&D, 1]

外键指向空记录

ID	Name	Department
93	Andrew	100

新增外键表字段

ID	Name	Department	DEPT_NAME
93	Andrew	100	

ID	Name	Department
93	Andrew	100

新增外键对象

ID	Name	Department	DEPT
93	Andrew	100	

14.3 外键

【例14.5】查询所有员工的姓名和部门名称，员工的部门不在部门表中的显示原值。员工表和部门表数据分别从文件Employee.txt 和Department.txt 中读取：

方法1：增加需要的外键表字段

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=file("Department.txt").import@t()	/从文件中导入部门表
3	=A1.join(Department, A2:ID, Name:DEPT_NAME)	/使用join函数，员工表按部门字段连接部门表，增加部门名称字段。
4	=A3.new(Name, if(DEPT_NAME,DEPT_NAME,Department):DEPT_NAME)	/创建序表，当部门名称为空时使用原值

方法2：外键对象产生在新增字段上

	A
3	=A1.join(Department, A2:ID, ~:DEPT)
4	=A3.new(Name,if(DEPT, DEPT.Name, Department):DEPT_NAME)

A4	Name	DEPT_NAME
	Rebecca	R&D
	Andrew	100
	…	…

注意

- 函数if(a,b,c) 中如果a非空且不是false返回b否则返回c。

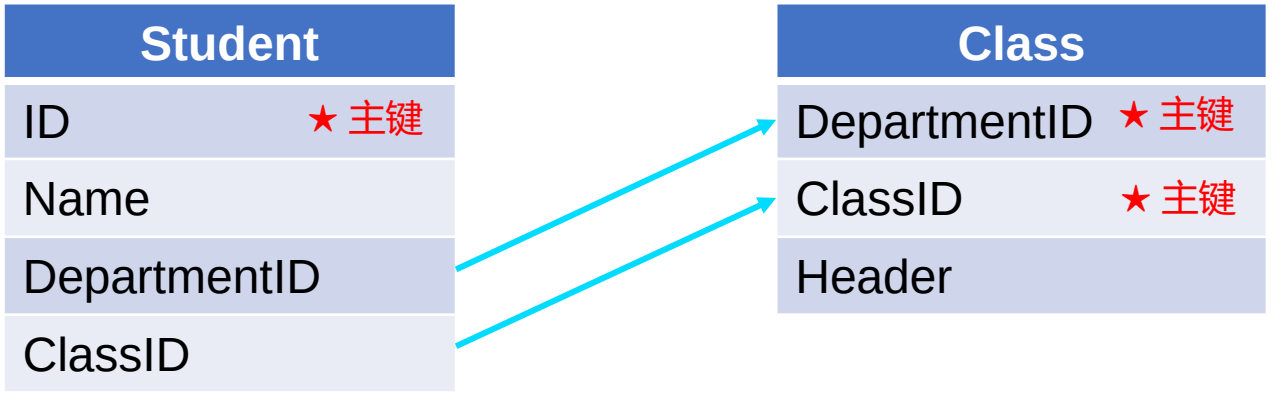
前面介绍过**主键**由表中的一个或多个字段组成。什么情况下需要多个字段联合作为主键呢?

Class	
DepartmentID	★ 主键
ClassID	★ 主键
Header	

DepartmentID	ClassID	Header
1	1	Sarah
1	2	Andrew
2	1	Madison
2	2	Emily

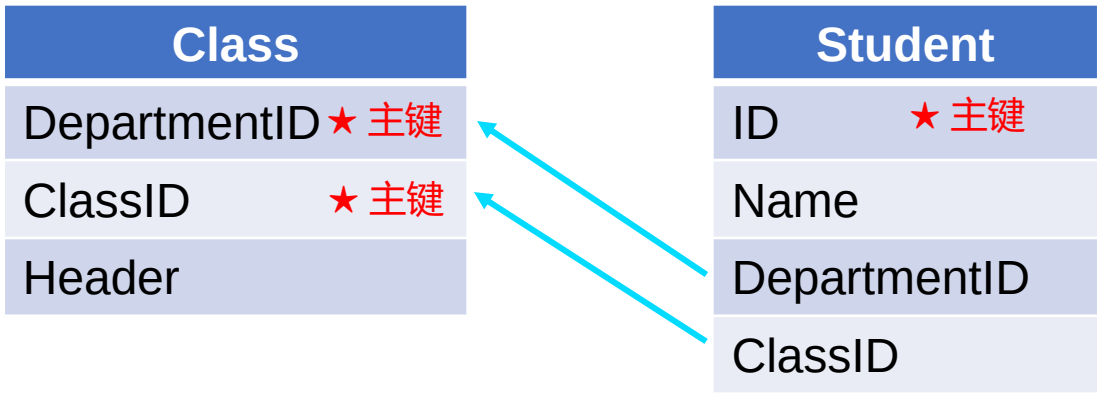
以班级表为例，数学系有一班和二班，计算机系也有一班和二班。专业ID和班级ID都不能单独的作为唯一性约束，只有专业ID和班级ID联合起来才能组成具有唯一约束的主键。

当外键表有**多字段主键**时，数据表就需要使用**多字段外键**联合起来与外键表的主键进行连接了。



上图中，学生的专业ID和班级ID组成了指向班级表的多字段外键。

【例14.6】查询每个学生的班主任。班级表和学生表数据分别从文件Class.txt 和Student.txt中读取：



【例14.6】查询每个学生的班主任。班级表和学生表数据分别从文件Class.txt 和Student.txt 中读取：

解题思路

使用多字段外键进行连接时，就不能在原字段上进行**外键对象化**了，可以在连接的同时新增字段保存外键表字段值。

学生表

ID	Name	DepartmentID	ClassID
1	Rebecca	1	1
2	Ashley	1	2

班级表

DepartmentID	ClassID	Header
1	1	Sarah
1	2	Andrew

增加外键映射字段

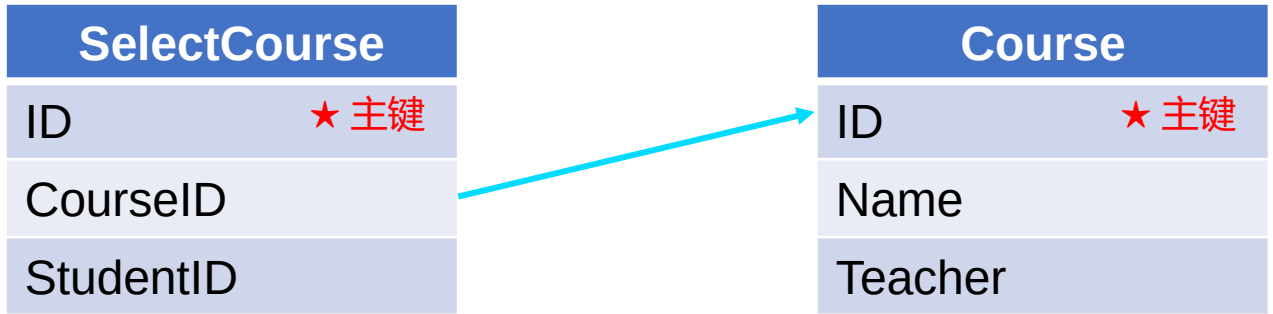
ID	Name	DepartmentID	ClassID	Header
1	Rebecca	1	1	Sarah
2	Ashley	1	2	Andrew

【例14.6】查询每个学生的班主任。班级表和学生表数据分别从文件Class.txt 和Student.txt 中读取：

	A	B
1	=file("Student.txt").import@t()	/从文件中导入学生表
2	=file("Class.txt").import@t()	/从文件中导入班级表
3	=A1.join(DepartmentID:ClassID, A2:DepartmentID:ClassID, Header)	/使用专业ID和班级ID字段连接学生表和班级表，连接时增加班级表的班主任字段到成绩表。
4	=A3.new(Name, Header)	/创建序表，返回学生姓名和班主任

A4	Name	Header
	Rebecca	Sarah
	Ashley	Sarah
	…	…

【例14.7】查询有多少学生选修了“Matlab”课程。课程表和选课表数据分别从文件Course.txt 和SelectCourse.txt 中读取：



【例14.7】查询有多少学生选修了“Matlab”课程。课程表和选课表数据分别从文件Course.txt和SelectCourse.txt 中读取：

解题思路

先在课程表中选出Matlab课程的记录，再进行连接过滤。

课程表

ID	Name	Teacher
~~1~~	~~Math~~	~~3~~
2	Matlab	18

过滤

ID	Name	Teacher
2	Matlab	18

选课表

ID	CourseID	StudentID
1	2	12
~~2~~	~~1~~	~~37~~

连接过滤

ID	CourseID	StudentID
1	2	12

【例14.7】查询有多少学生选修了“Matlab”课程。课程表和选课表数据分别从文件Course.txt和SelectCourse.txt 中读取：

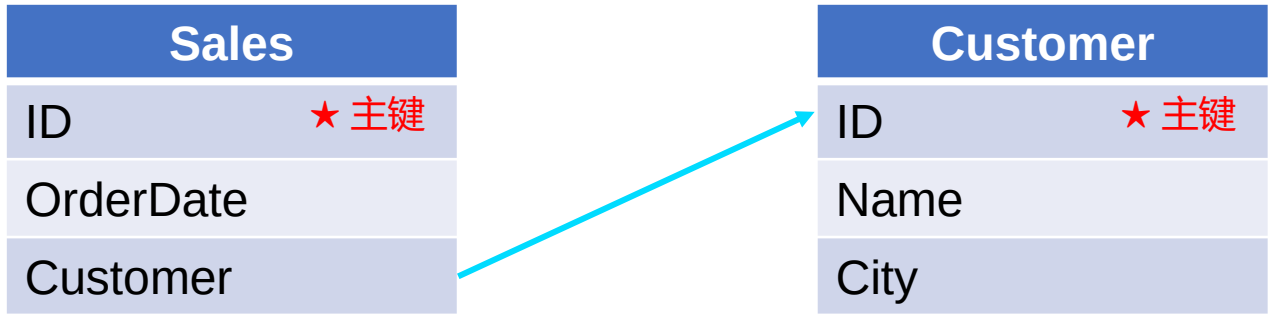
	A	B
1	=file("Course.txt").import@t()	/从文件中导入课程表
2	=file("SelectCourse.txt").import@t()	/从文件中导入选课表
3	=A1.select(Name:"Matlab")	/选出课程名称是Matlab的记录
4	=A2.switch@i(CourseID, A3:ID).count()	/使用switch函数的 i 选项，进行连接过滤。函数count用于计数。

A4
5

注意

- 函数**A.switch@i() / A.join@i()**用于连接过滤，连接时找不到对应值则删除该记录。
  A4可以替换为：**=A2.join@i(ID, A3:ID).count()**

【例14.8】查询2014年新增客户的销售情况（即销售表的客户ID不在客户表中的记录）。销售表和客户表数据分别从文件Sales.txt 和Customer.txt 中读取：



【例14.8】查询2014年新增客户的销售情况（即销售表的客户ID不在客户表中的记录）。销售表和客户表数据分别从文件Sales.txt 和Customer.txt 中读取：

解题思路

本题的思路是，在连接时只保留无匹配的记录。

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=file("Customer.txt").import@t()	/从文件中导入客户表
3	=A1.select(year(OrderDate)==2014)	/从销售表中选出2014年的记录
4	=A3.switch@d(Customer, A2:ID)	/使用switch函数的d选项，连接时只保留无匹配的记录。

A4

ID	OrderDate	Customer
10439	2014/02/07	1001
10504	2014/04/11	1002
…	…	…

注意

- 函数**A.switch@d(F, T:x)** 用于在连接时只保留无匹配的记录，此时不会把外键字段设置为null 。

中国行政区表如下，ID是行政区的编号，PARENT_ID是上级行政区的编号。比如唐山的上级行政区是河北省，同一个表的两个字段之间产生了关联：

ID	NAME	PARENT_ID
13	Hebei	1
…	…	…
1302	Tangshan	13

自连接是将自身表的一个镜像当作另一个表来对待，从而进行连接。常用于表中的两个不同的字段互相关联的情况。

【例14.9】查询每个机构的上级机构名称。组织机构表数据从文件Organization.txt 中读取：

ID	ORG_NAME	PARENT_ID
1	Head Office	0
2	Beijing Branch Office	1
3	Shanghai Branch Office	1
4	Chengdu Branch Office	1
5	Beijing R&D Center	2
6	Beijing Marketing Department	2
…	…	…

【例14.9】查询每个机构的上级机构名称。组织机构表数据从文件Organization.txt 中读取：

解题思路

组织机构表与自身进行外键连接，通过外键对象化，把上级机构字段转换成相应的机构记录。

ID	ORG_NAME	PARENT_ID
1	Head Office	0
2	Beijing Branch Office	1
…	…	…
6	Beijing Marketing Department	2

外键对象化

ID	ORG_NAME	PARENT_ID
1	Head Office	(null)
2	Beijing Branch Office	[1,Head Office,]
…	…	…
6	Beijing Marketing Department	[2,Beijing Branch Office,]

【例14.9】查询每个机构的上级机构名称。组织机构表数据从文件Organization.txt 中读取：

	A	B
1	=file("Organization.txt").import@t()	/从文件中导入组织机构表
2	=A1.switch(PARENT_ID, A1:ID)	/通过外键对象化，将上级机构ID字段转换成相应机构记录
3	=A2.new(ID, ORG_NAME, PARENT_ID.ORG_NAME:PARENT_NAME)	/创建序表，列出机构ID、机构名称和上级机构名称

A3

ID	ORG_NAME	PARENT_NAME
1	Head Office	(null)
2	Beijing Branch Office	Head Office
3	Shanghai Branch Office	Head Office
…	…	…

外键表最常用的解决方案是**外键对象化**。

外键对象化不可用的情况：

- 外键无对应记录。
- 多字段外键。

解决方案：

- 连接的同时新增字段保存外键表字段值。
- 连接的同时新增字段保存对应的外键表记录。

14.4 同维表和主子表

接下来介绍三种连接类型：**左连接**（**右连接**只是方向不同，不单独介绍了）、**内连接**和**全连接**。三种连接类型的区别如下：

- **左连接**
- **内连接**
- **全连接**

左连接也叫左外连接，以左表为基础将两表连接起来。结果会将左表所有的查询信息列出，而右表只列出条件与左表满足的部分。

订单表

ID	Customer	Date
10814	VICTE	2014/01/05
10815	SAVEA	2014/01/05

订单明细表

ID	ProductID	Price
10814	48	100
10814	48	300
10848	9	291

左连接

Order	Detail
[10814, VICTE, 2014/01/05]	[10814, 48, 100]
[10814, VICTE, 2014/01/05]	[10814, 48, 300]
[10815, SAVEA, 2014/01/05]	(null)

内连接也叫连接，会从结果表中删除与其他被连接表中没有匹配行的所有行。

订单表

ID	Customer	Date
10814	VICTE	2014/01/05
10815	SAVEA	2014/01/05

订单明细表

ID	ProductID	Price
10814	48	100
10814	48	300
10848	9	291

内连接

Order	Detail
[10814, VICTE, 2014/01/05]	[10814, 48, 100]
[10814, VICTE, 2014/01/05]	[10814, 48, 300]

全连接返回各表中的所有记录。当某记录在表中有匹配时，结果集包含该记录，否则置为空值。

订单表

ID	Customer	Date
10814	VICTE	2014/01/05
10815	SAVEA	2014/01/05

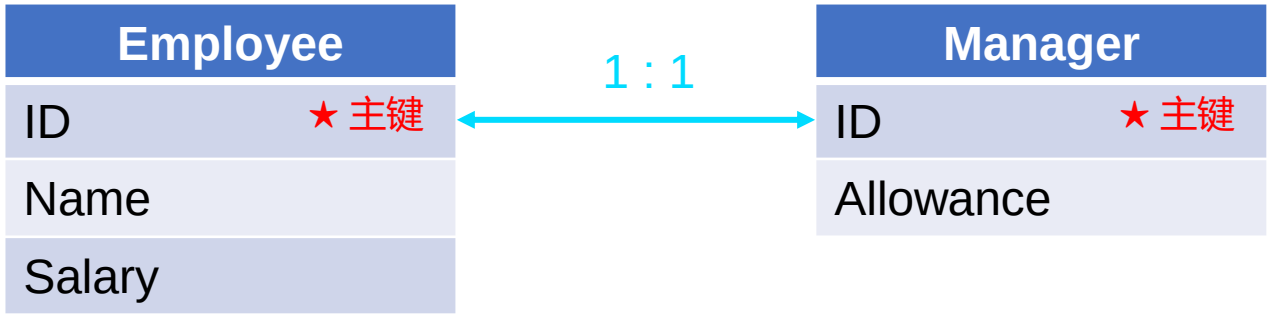
订单明细表

ID	ProductID	Price
10814	48	100
10814	48	300
10848	9	291

全连接

Order	Detail
[10814, VICTE, 2014/01/05]	[10814, 48, 100]
[10814, VICTE, 2014/01/05]	[10814, 48, 300]
[10815, SAVEA, 2014/01/05]	(null)
(null)	[10848, 9, 291]

【例14.10】要统计所有员工（包括经理）的总收入（加上津贴）。员工表和经理表数据分别从文件Employee.txt 和Manager.txt 中读取：



【例14.10】要统计所有员工（包括经理）的总收入（加上津贴）。员工表和经理表数据分别从文件Employee.txt 和Manager.txt 中读取：

解题思路

员工表和经理表是**同维表**关系，可以直接按照**主键**进行连接。因为要查询的是全体员工，把员工表作为左表进行左连接。

员工表

ID	Name	Salary
1	Rebecca	7000

经理表

ID	Allowance
1	8000

用主键ID左连接

Employee	Manager
[1, Rebecca, 7000]	[1, 8000]

14.4 同维表和主子表

【例14.10】要统计所有员工（包括经理）的总收入（加上津贴）。员工表和经理表数据分别从文件Employee.txt 和Manager.txt 中读取：

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=file("Manager.txt").import@t().keys(ID)	/从文件中导入经理表
3	=create(Employee, Manager)	/创建序表，以员工和经理为字段
4	>A1.(A3.insert(0, A1.~, A2.find(A1.~.ID)))	/循环员工表，根据员工编号在经理表中查找对应记录，并将员工记录和对应的经理记录追加到A3的序表中
5	=A3.new(Employee.ID: ID, Employee.Name:Name, Employee.Salary+Manager.Allowance:Income)	/创建序表，统计员工的总收入（工资+津贴）

A5	ID	Name	Income
	1	Rebecca	15000
	2	Ashley	22000
	…	…	…

14.4 同维表和主子表

【例14.10】要统计所有员工（包括经理）的总收入（加上津贴）。员工表和经理表数据分别从文件Employee.txt 和Manager.txt 中读取：

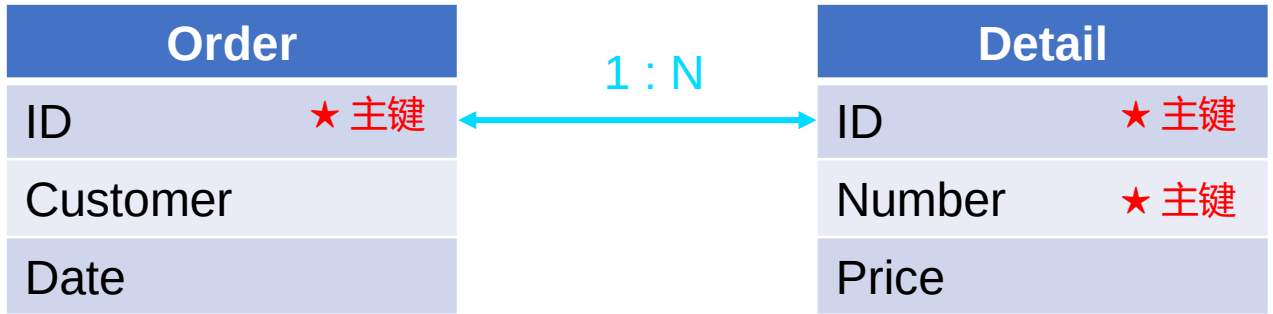
解题思路

可以使用join函数来简化连接的过程。函数**join(Ai:Fi; ...)**用于将序表按照指定字段进行连接。结果集是以集合成员构成的二元组为成员。**join@1()**表示针对第一个序表左连接。

	A	B
1	=file("Employee.txt").import@t()	/从文件中导入员工表
2	=file("Manager.txt").import@t()	/从文件中导入经理表
3	=join@1(A1:Employee, ID; A2:Manager, ID)	/使用join@1函数按照主键ID连接员工表和经理表，针对员工表左连接
4	=A3.new(Employee.ID: ID, Employee.Name:Name, Employee.Salary+Manager.Allowance:Income)	/创建序表，统计员工的总收入（工资+津贴）

A4	ID	Name	Income
	30	Joseph	18000
	400	Joshua	16000
	…	…	…

【例14.11】统计每个订单的总金额。订单表和订单明细表数据分别从文件Order.txt 和Detail.txt 中读取：



【例14.11】统计每个订单的总金额。订单表和订单明细表数据分别从文件Order.txt 和Detail.txt 中读取：

解题思路

单个**主子表**时，可以通过子表的**部分主键**与主表**主键**进行连接。

订单表

ID	Customer	Date
10248	VINET	2012-07-04

订单明细表

ID	Number	Price
10248	1	238
10248	2	475

按主表主键订单ID连接

Order	Detail
[10248,VINET,2012-07-04]	[10248,1,238]
[10248,VINET,2012-07-04]	[10248,2,475]

分组汇总

ID	Customer	Amount
10248	VINET	713

【例14.11】统计每个订单的总金额。订单表和订单明细表数据分别从文件Order.txt 和Detail.txt 中读取：

	A	B
1	=file("Order.txt").import@t()	/从文件中导入订单表
2	=file("Detail.txt").import@t()	/从文件中导入订单明细表
3	=join(A1:Order,ID; A2:Detail,ID)	/使用join函数按照订单ID连接订单表和订单明细表
4	=A3.groups(Order.ID:ID, Order.Customer:Customer; sum(Detail.Price):Amount)	/统计每个订单的总金额

A4	ID	Customer	Amount
	10814	VICTE	816.0
	10848	CONSH	3201.0
	…	…	…

注意

- 如果订单和订单明细针对id都是有序的，我们可以使用函数**join@m()**进行有序归并，从而提高join的速度与性能。

【例14.12】比较2014年和2015年各种产品的销售数量。销售表数据从文件Sales2014.txt 和Sales2015.txt 中读取：

OrderID	Customer	Product	Date	Amount
10400	VINET	Cheese	2014/01/01	2440.0
10401	TOMSP	Coffee	2014/01/01	1863.4
10402	HANAR	Milk	2014/01/02	1813.0
…	…	…	…	…

14.4 同维表和主子表

【例14.12】比较2014年和2015年各种产品的销售额。销售表数据从文件Sales2014.txt 和Sales2015.txt 中读取：

解题思路

不论产品在哪个年份中有销售都应该列出，所以本例应该使用**全连接**。可以先统计每个年份的各产品销量，不同年份的产品销量是**同维表**关系，可以直接进行连接。

	A	B
1	=file("Sales2014.txt").import@t()	/从文件中导入2014年销售表
2	=file("Sales2015.txt").import@t()	/从文件中导入2015年销售表
3	=A1.groups(Product; sum(Amount):Amount)	/分组汇总2014年各产品的销售额
4	=A2.groups(Product; sum(Amount):Amount)	/分组汇总2015年各产品的销售额
5	=join@f(A3:A2014, Product; A4:A2015, Product)	/对2014和2015年的各产品销售额进行全连接
6	=A5.new(ifn(A2014.Product, A2015.Product):Product, A2014.Amount:A2014, A2015.Amount:A2015)	/创建序表，以产品、2014年销售额和2015年销售额为字段。

A6

Product	C2014	C2015
Biscuits	8931.47	266.0
Shrimp eggs	4345.05	6726.05
…	…	…

注意

- 函数**join@f(Ai:Fi; …)** 用于全连接。

【例14.13】订单表还有一个子表用于记录回款情况。我们现在想知道哪些订单还没有回款，也就是累计回款金额小于订单总金额的订单。订单回款表数据从文件Payment.txt中读取：

【例14.13】订单表还有一个子表用于记录回款情况。我们现在想知道哪些订单还没有回款，也就是累计回款金额小于订单总金额的订单。订单回款表数据从文件Payment.txt中读取：

解题思路

简单地把这三个表 JOIN 起来是不对的，订单明细和订单回款表会发生多对多的关系。子表按 ID 分组后，就变成了具有唯一性的事实上的主键，多表之间就可以直接用**主键**进行连接了。

订单明细表

ID	Number	Price
10248	1	238
10248	2	475

子表按ID分组

Member
[[10248,1,238],[10248,2,475]]

订单回款表

ID	Date	Amount
10248	2012-07-26	238
10248	2012-08-15	475

子表按ID分组

Member
[[10248,2012-07-26,238],[10248,2012-08-15,475]]

订单表

ID	Customer	Date
10248	VINET	2012-07-04

多表通过ID连接

Order	Detail	Payment
[10248,VINET,2012-07-04]	[[10248,1,238],[10248,2,475]]	[[10248,2012-07-26,238], …]

14.4 同维表和主子表

【例14.13】订单表还有一个子表用于记录回款情况。我们现在想知道哪些订单还没有回款，也就是累计回款金额小于订单总金额的订单。订单回款表数据从文件Payment.txt中读取：

	A	B
1	=file("Order.txt").import@t()	/从文件中导入订单表
2	=file("Detail.txt").import@t()	/从文件中导入订单明细表
3	=file("Payment.txt").import@t()	/从文件中导入订单回款表
4	=A2.groups(ID; sum(Price):a)	/订单明细表按ID分组汇总，统计销售额
5	=A3.groups(ID; sum(Amount):p)	/订单回款表按ID分组汇总，统计回款金额
6	=join(A1:Order,ID; A4:Detail,ID; A5:Payment,ID)	/使用join函数按照订单ID连接多表
7	=A6.new(Order.ID:ID, Detail.a:Amount, Payment.p:Payment)	/创建序表，统计每个订单的总金额和总回款
8	=A7.select(Payment<Amount)	/选出回款金额小于总金额的记录，即没有完成回款的记录。

A8

ID	Amount	Payment
10848	3201.0	2400.75
…	…	…

注意

- 多个主子表的情况，子表先按照主表主键进行分组，此时该主键就是事实上的主键了。然后多个表按主表主键进行连接即可。

14.5 交叉连接与非等值连接

14.5.1 交叉连接

学生表与科目表进行连接，每一个学生记录都与所有的科目记录组合成一条新记录。所有的学生记录与科目记录一一配对，结果集的长度是学生表的长度与科目表的长度的乘积。

学生表

STU_ID	STU_NAME
1	Rudy
2	Simon

×

科目表

SUB_ID	SUB_NAME
1	Math
2	PE

=

Student	Subject
[1,Rudy]	[1,Math]
[1,Rudy]	[2,PE]
[2,Simon]	[1,Math]
[2,Simon]	[2,PE]

上面这种连接方式，就是交叉连接。表 X 与表 Y 交叉连接的结果集中，第一个字段是 X 的成员，而第二个字段是 Y 的所有可能有序对的其中一个成员。

14.5.1 交叉连接

【例14.14】求两个矩阵的乘积。两个矩阵表数据分别从文件 MatrixA.txt 和 MatrixB.txt 中读取。

矩阵表的数据结构如下：

row	col	value
1	1	1
1	2	2
1	3	3
…	…	…

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

本例的数学公式如下：

$$C = AB = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix} = \begin{pmatrix} 1\times1+2\times2+3\times3 & 1\times4+2\times5+3\times6 \\ 4\times1+5\times2+6\times3 & 4\times4+5\times5+6\times6 \end{pmatrix} = \begin{pmatrix} 14 & 32 \\ 32 & 77 \end{pmatrix}$$

14.5.1 交叉连接

【例14.14】求两个矩阵的乘积。两个矩阵表数据分别从文件 MatrixA.txt 和 MatrixB.txt 中读取。

解题思路

第一步，将两个矩阵表叉乘，叉乘的同时进行过滤，选出A的列号等于B的行号的记录。
第二步，对行列号相同的值的乘积求和。

第一步：

row	col	value
1	1	1
1	2	2
1	3	3
2	1	4
2	2	5
2	3	6

×

row	col	value
1	1	1
1	2	4
2	1	2
2	2	5
3	1	3
3	2	6

=

A	B
[1, 1, 1]	[1, 1, 1]
[1, 1, 1]	[1, 2, 4]
~~[1, 1, 1]~~	~~[2, 1, 2]~~
~~[1, 1, 1]~~	~~[2, 2, 5]~~
~~[1, 1, 1]~~	~~[3, 1, 3]~~
…	…

第二步：

row	col	sum(A.value*B.value)
1	1	14
1	2	32
2	1	32
2	2	77

14.5.1 交叉连接

【例14.14】求两个矩阵的乘积。两个矩阵表数据分别从文件 MatrixA.txt 和 MatrixB.txt 中读取。

	A	B
1	=file("MatrixA.txt").import@t()	/从文件中导入矩阵表A
2	=file("MatrixB.txt").import@t()	/从文件中导入矩阵表B
3	=xjoin(A1:A; A2:B, A.col==B.row)	/使用xjoin函数进行交叉连接，在连接的同时按照条件过滤
4	=A3.groups(A.row:row, B.col:col; sum(A.value * B.value):value)	/分组汇总相同行列号的值的乘积之和

A4	row	col	value
	1	1	14
	1	2	32
	2	1	32
	2	2	77

$$\begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}$$

注意

- 函数**xjoin(A_i:F_i,x_i; ...)**用于对多表A_i进行交叉连接，有条件x_i时，会在连接的同时过滤。

14.5.2 非等值连接

连接条件不是等值比较时，称为**非等值连接**。非等值连接在实际应用中比较少见。

例如查询员工的收入等级：

员工表

EID	NAME	SALARY
1	Simon	12000
2	Rudy	7000

收入等级表

BAND	START	END
Low income	0	8000
Middle income	8000	15000
High income	15000	30000

员工表和收入等级表之间是非等值连接：当员工收入在某收入等级的上下限之间时，即为员工的收入等级。

14.5.2 非等值连接

【例14.15】查询社区居民所处的年龄段。社区居民表和年龄区间表分别从文件 Community.txt 和 AgeGroup.txt 中读取。

社区居民表

ID	Name	Age
1	David	28
2	Daniel	15
3	Andrew	65
4	Rudy	(null)

年龄区间表

Group	Start	End
Children	0	15
Youth	16	40
Middle	41	60
Old	61	100

14.5.2 非等值连接

【例14.15】查询社区居民所处的年龄段。社区居民表和年龄区间表分别从文件 Community.txt 和 AgeGroup.txt 中读取。

解题思路

两表叉乘，同时按照起始和结束年龄比较出所处年龄段。

ID	Name	Age
1	David	28
2	Daniel	15
3	Andrew	65
…	…	…

×

Group	Start	End
Children	0	15
Youth	16	40
Middle	41	60
Old	61	100

=

Person	Age
~~[1,David,28]~~	~~[Children,0,15]~~
[1,David,28]	[Youth,16,40]
~~[1,David,28]~~	~~[Middle,41,60]~~
~~[1,David,28]~~	~~[Old,61,100]~~
…	…

14.5.2 非等值连接

【例14.15】查询社区居民所处的年龄段。社区居民表和年龄区间表分别从文件 Community.txt 和 AgeGroup.txt 中读取。

	A	B
1	=file("Community.txt").import@t()	/从文件中导入社区居民表
2	=file("AgeGroup.txt").import@t()	/从文件中导入年龄区间表
3	=xjoin@1(A1:Person; A2:Age, A2.Start<=Person.Age && A2.End>=Person.Age)	/两表进行交叉连接，同时选出年龄在相应的年龄区间的记录
4	=A3.new(Person.ID:ID, Person.Name:Name, Person.Age:Age,Age.Group:Group)	/生成新序表

A4	ID	Name	Age	Group
	1	David	28	Youth
	2	Daniel	15	Children
	3	Andrew	65	Old
	4	Rudy	(null)	(null)

注意

- 函数**xjoin@1()** 在交叉连接时使用左连接，当年龄区间无法匹配时保留社区居民记录，年龄段置为空。

14.6 习题

14.6 习题

【习题14.1】查询OrderID是55080325的销售金额。销售表数据来自Sales.txt文件：

OrderID	Customer	OrderDate	Amount
81182311	VINET	2013/07/04	2440.0
98807954	TOMSP	2013/07/05	1863.4
…	…	…	…
55080325	WHITC	2015/05/01	928.75
…	…	…	…

14.6 习题

【习题14.1】查询ID是55080325的销售金额。销售表数据来自Sales.txt文件：

OrderID	Customer	OrderDate	Amount
81182311	VINET	2013/07/04	2440.0
98807954	TOMSP	2013/07/05	1863.4
…	…	…	…
55080325	WHITC	2015/05/01	928.75
…	…	…	…

解题思路

可以利用表的**主键值**查找到指定的记录。

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入销售表
2	=A1.keys(OrderID)	/设置销售表的主键为OrderID
3	=A2.find(55080325)	/使用函数 find 查找主键为55080325的销售记录
4	=A3.Amount	/从销售记录中取出销售额

A4
Value
928.75

14.6 习题

【习题14.2】查询每个学生姓名及其数学分数。成绩表和学生表数据分别从文件Score.txt 和Student.txt 中读取：

14.6 习题

【习题14.2】查询每个学生姓名及其数学分数。成绩表和学生表数据分别从文件Score.txt 和Student.txt 中读取：

解题思路

外键表可以使用**外键对象化**，将成绩表的学生字段转换成学生表中的对应记录。

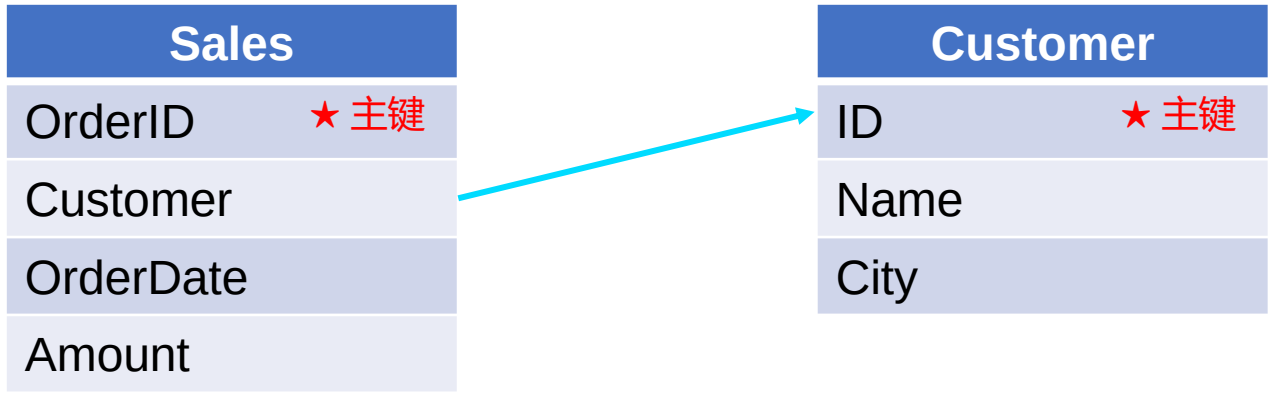
	A	B
1	=file("Score.txt").import@t()	/从文件中导入成绩表
2	=file("Student.txt").import@t()	/从文件中导入学生表
3	=A1.select(SUBJECT:"Math")	/成绩表中选出科目是数学的记录
4	=A3.switch(STUDENTID, A2:ID)	/使用switch函数，将学生字段值转换为学生表对应的记录
5	=A4.new(STUDENTID.Name, SCORE)	/创建序表，生成每个学生的姓名和数学分数

A5

ID	Name	SCORE
1	Rebecca	77
…	…	…

14.6 习题

【习题14.3】查询每个销售记录的客户名称、日期和销售额，并导出到CustomerSales.txt文件。某些客户可能不在客户表中。销售表和客户表数据分别从文件Sales.txt 和Customer.txt 中读取：



14.6 习题

【习题14.3】查询每个销售记录的客户名称、日期和销售额，并导出到CustomerSales.txt文件。某些客户可能不在客户表中。销售表和客户表数据分别从文件Sales.txt 和Customer.txt 中读取：

解题思路

外键指向空记录的情况，可以增加需要的外键表字段。

	A	B
1	=file("Sales.txt").import@t()	/从文件中导入员工表
2	=file("Customer.txt").import@t()	/从文件中导入岗位表
3	=A1.join(Customer, A2:ID, Name:CustomerName)	/使用join函数连接两表，销售表增加客户表的客户名称字段。
4	=A3.new(CustomerName, OrderDate, Amount)	/创建序表，以客户名称、日期和销售额作为字段
5	>file("CustomerSales.txt").export@t(A4)	/导出A4创建的序表到文件CustomerSales.txt

CustomerSales.txt 如下：

```
CustomerName        OrderDate Amount
Shantai     2013/07/04          2440.0
Shiyi       2013/07/05          1863.4
...
```

14.6 习题

【习题14.4】查询每个学生的姓名和家庭住址。学生表和学生联系表数据分别从文件Student.txt和StudentContacts.txt 中读取：

14.6 习题

【习题14.4】查询每个学生的姓名和家庭住址。学生表和学生联系表数据分别从文件Student.txt和StudentContacts.txt 中读取：

解题思路

同维表的关系是相等的，可以直接连接两个表。

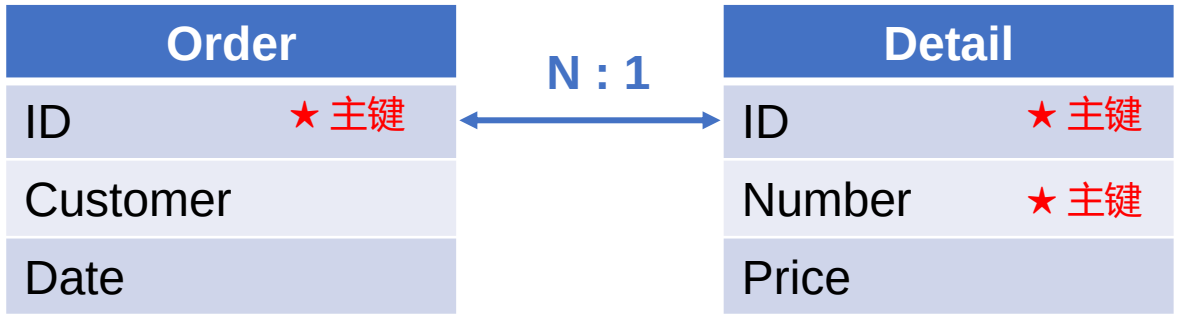
	A	B
1	=file("Student.txt").import@t()	/从文件中导入学生表
2	=file("StudentContacts.txt").import@t()	/从文件中导入学生联系表
3	=join(A1:Student, ID; A2:Contacts, ID)	/使用join函数按照主键学生ID连接学生表和学生联系表
4	=A3.new(Student.Name:Name, Contacts.Address:Address)	/创建序表，查询学生的姓名和家庭住址

A4

ID	Name	Address
1	Rebecca	124 Guangming North Road
…	…	…

14.6 习题

【习题14.5】统计2014年每月的销售额。订单表和订单明细表数据分别从文件Order.txt 和Detail.txt 中读取：



14.6 习题

【习题14.5】统计2014年每月的销售额。订单表和订单明细表数据分别从文件Order.txt 和Detail.txt 中读取：

解题思路

单个的**主子表**关系，可以直接按主键共有字段进行连接。

ID	Customer	Date
10814	VICTE	2014-01-05

ID	Number	Price
10814	1	102.0
10814	2	102.0
…	…	…

按共有字段订单ID连接

Order	Detail
[10814,VICTE,2014-01-05]	[10814,1,102.0]
[10814,VICTE,2014-01-05]	[10814,2,102.0]
…	…

14.6 习题

【习题14.5】统计2014年每月的销售额。订单表和订单明细表数据分别从文件Order.txt 和Detail.txt 中读取：

	A	B
1	=file("Order.txt").import@t()	/从文件中导入订单表
2	=A1.select(year(Date)==2014)	/从订单表中选出2014年的记录
3	=file("Detail.txt").import@t()	/从文件中导入订单明细表
4	=join(A2:Order,ID; A3:Detail,ID)	/使用join函数按共有字段订单ID连接订单和订单明细表
5	=A4.groups(month(Order.Date):Month; sum(Detail.Price):Amount)	/分组汇总每月订单的总金额

A5

Month	Amount
1	4017.0
2	31841.0
…	…

14.6 习题

【习题14.6】查询总销售额超过一万的客户名称。各表数据来自同名txt文件，表间关系如下：

14.6 习题

【习题14.6】查询总销售额超过一万的客户名称。各表数据来自同名txt文件，表间关系如下：

解题思路

本例中有外键表关系，也有单个的主子表关系。可以先进行**外键对象化**，再连接**主子表**。

	A	B
1	=file("Order.txt").import@t()	/从文件中导入订单表
2	=file("Detail.txt").import@t()	/从文件中导入订单明细表
3	=file("Customer.txt").import@t()	/从文件中导入客户表
4	=A1.switch(Customer, A3:ID)	/使用switch函数，将订单表的客户ID转换成对应记录
5	=join(A4:Order,ID; A2:Detail,ID)	/使用join函数按订单ID连接订单和订单明细表
6	=A5.groups(Order.Customer.Name:Name; sum(Detail.Price):Amount)	/分组汇总每个订单的总金额
7	=A6.select(Amount>10000)	/选出订单总金额超过10000的记录

A7

Name	Amount
ALFKI	39661.0
…	…

14.6 习题

【习题14.7】查询2014年没有全部发货的订单。各表数据来自同名txt文件，表间关系如下：

14.6 习题

【习题14.7】查询2014年没有全部发货的订单。各表数据来自同名txt文件，表间关系如下：

解题思路

本例中有多个主子表关系，子表先按ID 进行分组，ID就成了事实上的主键。然后多表之间就是**同维表**关系了，可以直接用ID连接。

订单明细表

ID	Number	Count
10814	1	8
10814	2	24

子表按ID分组

Member
[[10814,1,8],[10248,2,24]]

订单发货表

ID	Date	Count
10814	2014/01/08	8
10814	2014/01/13	24

子表按ID分组

Member
[[10248,2014/01/08,8],[10248,2012/08/13,24]]

订单表

ID	Customer	Date
10814	VICTE	2014/01/05

多表通过ID连接

Order	Detail	Delivery
[10248,VICTE,2014/01/05]	[[10814,1,8],[10248,2,24]]	[[10248,2014/01/08,8],[10248,2012/08/13,24]]

14.6 习题

【习题14.7】查询2014年没有全部发货的订单。各表数据来自同名txt文件，表间关系如下：

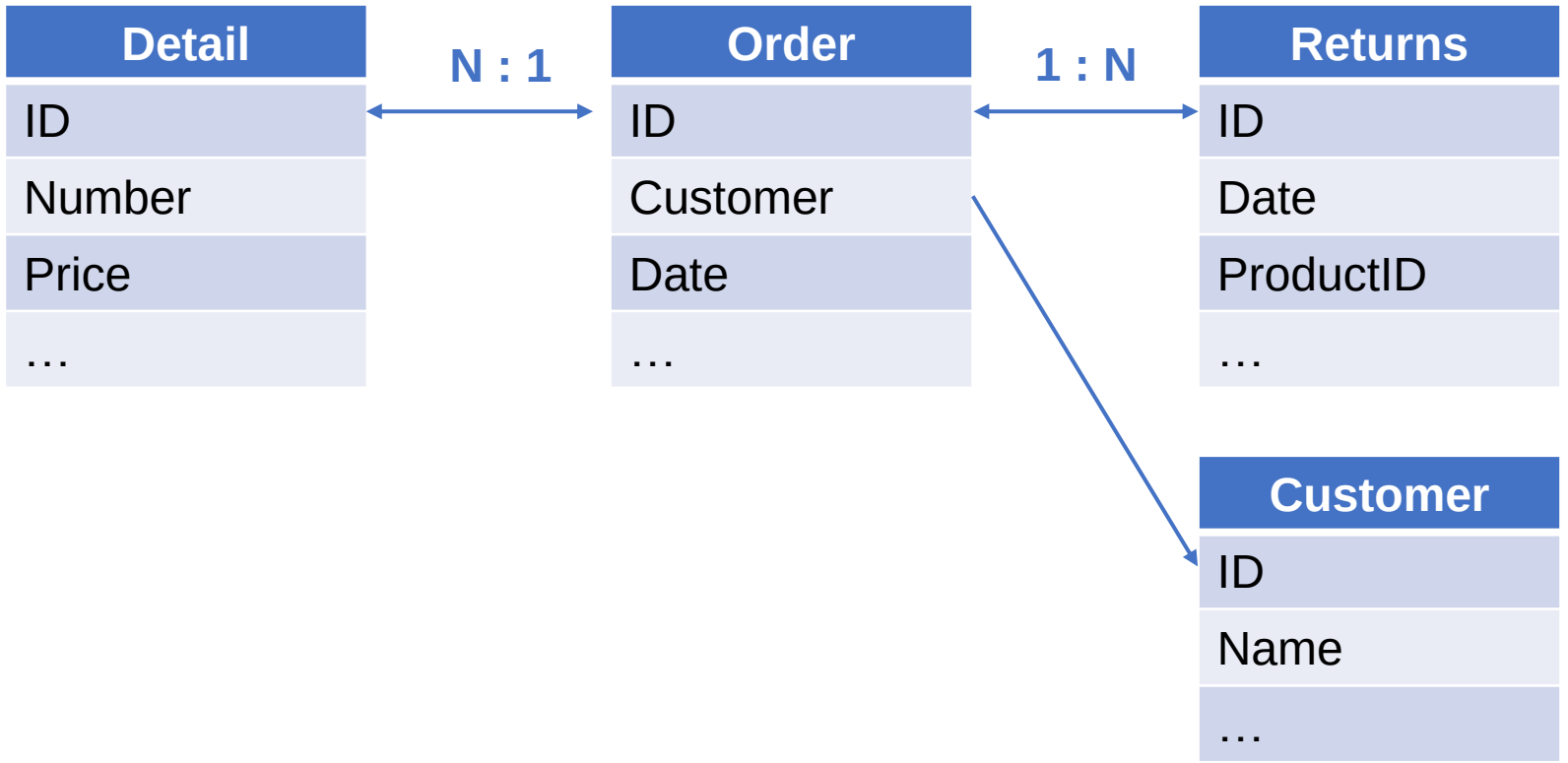
	A	B
1	=file("Order.txt").import@t()	/从文件中导入订单表
2	=file("Detail.txt").import@t()	/从文件中导入订单明细表
3	=file("Delivery.txt").import@t()	/从文件中导入订单发货表
4	=A2.group(ID)	/订单明细表按ID分组
5	=A3.group(ID)	/订单发货表按ID分组
6	=join(A1:Order,ID; A4:Detail,ID; A5:Delivery,ID)	/使用join函数按照订单ID连接多表
7	=A6.new(Order.ID:ID, Detail.sum(Count):Count, Delivery.sum(Count):DeliveryCount)	/创建序表，统计每个订单的货物数量和发货数量
8	=A7.select(DeliveryCount<Count)	/选出发货数量小于订单货物数量的记录，即没有完成发货的记录。

A8

ID	Count	DeliveryCount
10979	150	90
…	…	…

14.6 习题

【习题14.8】查询2015年各客户的退款订单数。各表数据来自同名txt文件，表间关系如下：



14.6 习题

【习题14.8】查询2015年各客户的退款订单数。各表数据来自同名txt文件，表间关系如下：

解题思路

本例是外键表、同维表和主子表混合的例子。按前面章节学习的方法：先进行外键对象化，再把子表按主表主键分组，最后多表都是同维表关系了直接用主键连接。

订单表

ID	Customer	Date
11087	ANTON	2015/02/03

外键对象化

ID	Customer	Date
11087	[ANTON, …]	2015/02/03

订单明细表

ID	Number	Price
11087	1	124.0
11087	2	31.0

子表按ID分组

Member
[[11087,1,124.0],[11087,2,31.0]]

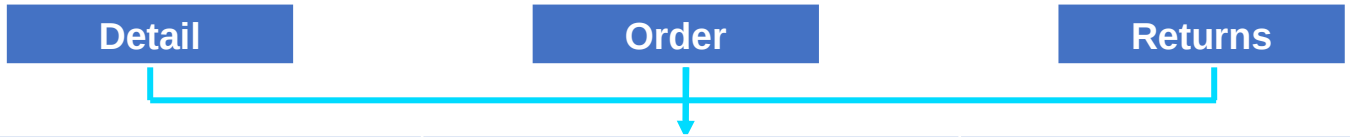
订单退款表

ID	Date	ProductID
11087	2015/02/13	77

子表按ID分组

Member
[[11087,2015/02/13,77]]

多表按ID连接



Detail	Order	Returns
[[11087,1,124.0],[11087,2,31.0]]	[11087,[ANTON,…], 2015/02/03]	[[11087,2015/02/13,77]]

14.6 习题

【习题14.8】查询2015年各客户的退款订单数。各表数据来自同名txt文件，表间关系如下：

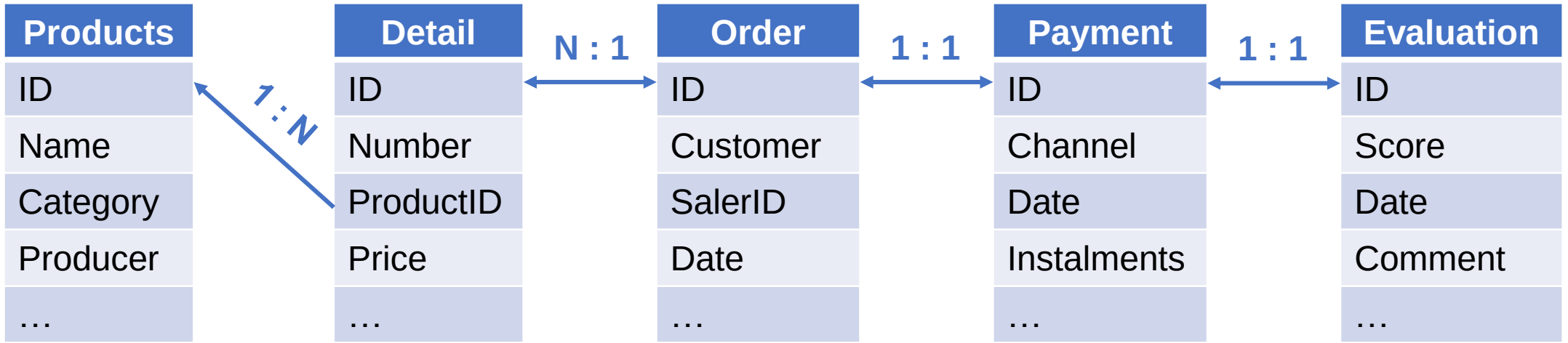
	A	B
1	=file("Order.txt").import@t()	/从文件中导入订单表
2	=file("Detail.txt").import@t()	/从文件中导入订单明细表
3	=file("Returns.txt").import@t()	/从文件中导入订单退货表
4	=file("Customer.txt").import@t()	/从文件中导入客户表
5	=A1.select(year(Date)==2015).switch(Customer, A4:ID)	/从订单表中选出2015年的记录，将客户外键对象化
6	=A2.group(ID)	/订单明细表按ID分组
7	=A3.group(ID)	/订单退款表按ID分组
8	=join(A5:Order, ID;A6:Detail, ID;A7:Returns, ID)	/使用join函数按照订单ID连接多表
9	=A8.groups(Order.Customer.Name; count(~):Count)	/使用groups函数分组汇总每个客户的退款订单数

A9

Name	Count
Guoding Co., Ltd	1
…	…

14.6 习题

【习题14.9】查询商品名称中包含“water”，在2014年1月份下单，订单总金额大于200元，没有使用分期，得到5星好评的订单信息。各表数据来自同名txt文件，表间关系如下：



14.6 习题

【习题14.9】查询商品名称中包含“water”，在2014年1月份下单，订单总金额大于200元，没有使用分期，得到5星好评的订单信息。各表数据来自同名txt文件，表间关系如下：

解题思路

这个例子中既包含同维表和主子表，也有外键表。需要把问题拆开来解决：先进行外键对象化，再把子表按主表主键分组，最后多表都是同维表关系了直接用主键连接。

ID	Number	ProductID	Price
10248	1	1	238

外键对象化

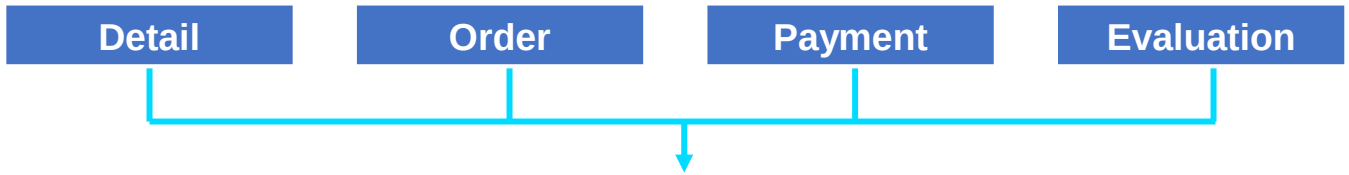
ID	Number	ProductID	Price
10248	1	[1,cake,3,12]	238

ID	Number	ProductID	Price
10248	1	[1,cake,3,12]	238
10248	2	[2,apple,3,5]	475

子表按ID分组

Member
[[10248,1,[1,cake,3,12],238],[10248,2,[2, …], 475]]

多表按ID连接



Detail	Order	Payment	Evaluation
[[10248,1,[1,cake,3,12],238], […]]	[10248,VINET, 2012-07-04]	[10248,3,2012-07-04,0]	[10248,5,2012-07-16,"Good"]

14.6 习题

【习题14.9】查询商品名称中包含“water”，在2014年1月份下单，订单总金额大于200元，没有使用分期，得到5星好评的订单信息。各表数据来自同名txt文件，表间关系如下：

	A	B	C
1	=file("Products.txt").import@t()	=A1.select(like(Name, "*water*"))	/从文件中导入产品表，选出产品名包含water的记录
2	=file("Detail.txt").import@t()	=A2.switch@i(ProductID, A1:ID)	/从文件中导入订单明细表。订单明细表对产品ID字段外键对象化，转换成产品表相应记录，并删除不匹配的记录。
3	=B2.group(ID)	=A3.select(sum(Price)>200)	/订单明细表按订单ID分组，选出订单总金额超过200的记录
4	=file("Order.txt").import@t()	=A4.select(year(Date)==2014 && month(Date)==1)	/从文件中导入订单表，选出2014年1月的记录
5	=file("Payment.txt").import@t()	=A5.select(Instalments==0)	/从文件中导入订单回款表，选出没有使用分期的记录
6	=file("Evaluation.txt").import@t()	=A6.select(Score==5)	/从文件中导入订单评价表，选出5星好评的记录
7	=join(B3:Detail,ID; B4:Order,ID; B5:Payment, ID; B6:Evaluation,ID)		/使用join函数按照订单ID连接多表
8	=A7.new(Order.ID:ID, Order.Customer:Customer, Order.Date:Date, Detail.sum(Price):Amount)		/创建序表，生成订单信息

A8

ID	Customer	Date	Amount
10848	CONSH	2014/01/23	3201.0

第15章 图形绘制

15.1 画布

- 画布
- 图元

15.2 物理坐标

- 绝对坐标
- 相对坐标

15.3 逻辑坐标

- 数值轴
- 枚举轴
- 日期轴

15.4 更多逻辑映射

- 极坐标系
- 组合坐标系
- 图例

15.5 成品图

- 柱状图
- 双轴图

15.6 习题

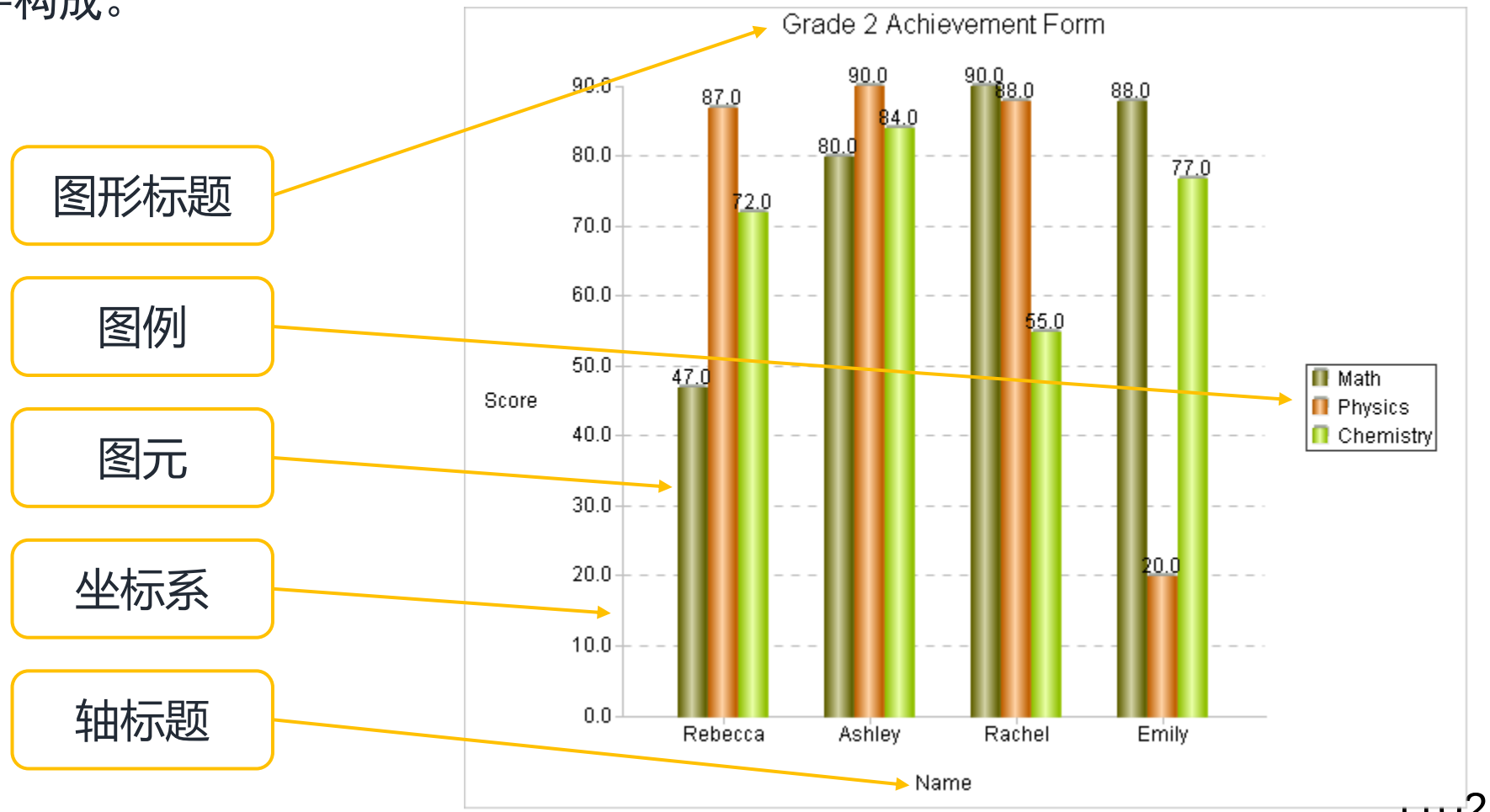
- 三角函数
- 多重组合
- K线图

15.1 画布

15.1.1 画布

绘图需要一个空白区域，在计算机上一般都是一个矩形，而要画的图则由这个区域中的点构成。这个区域，我们就称为画布。

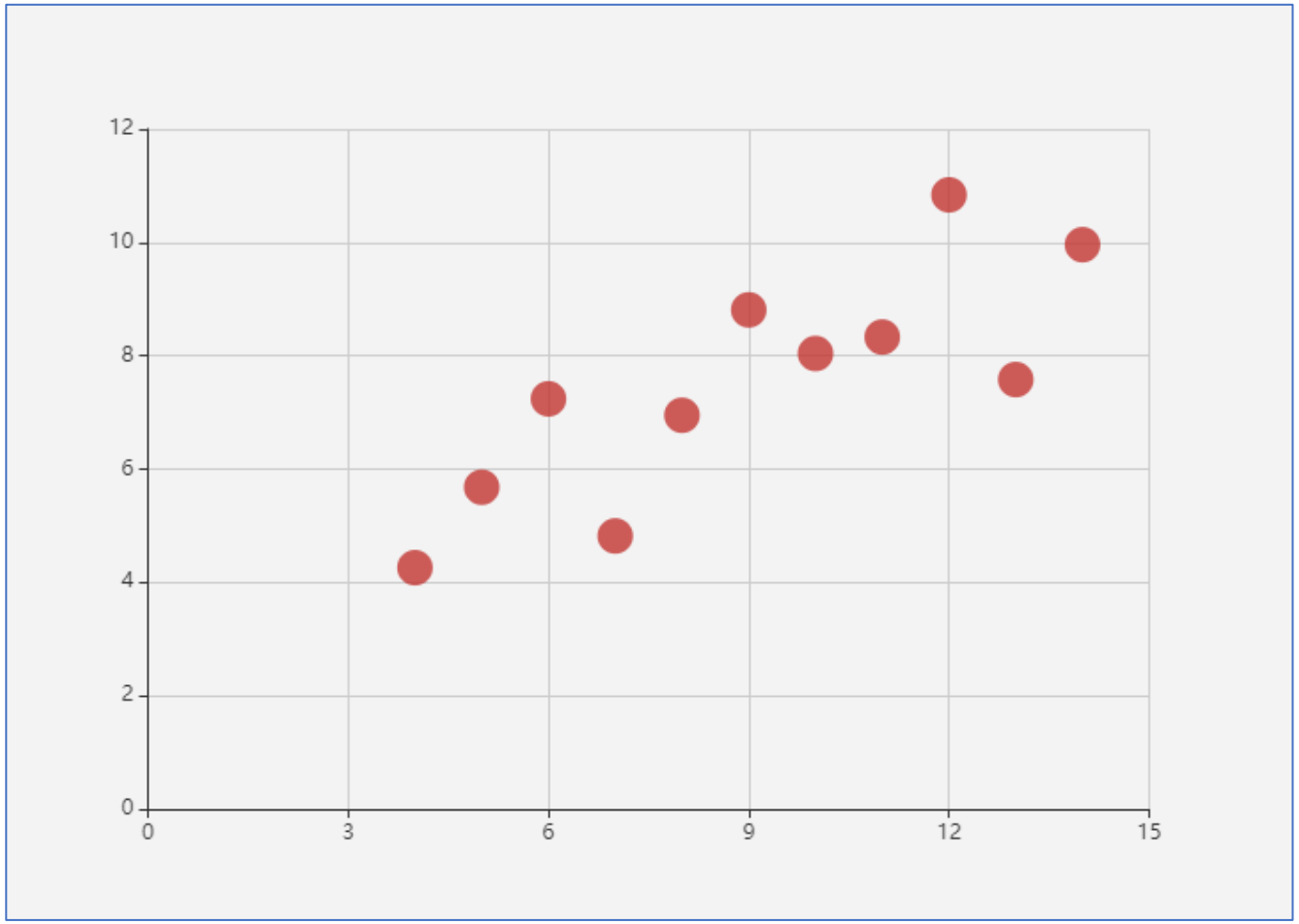
画布中图形由一些部件构成。



15.1.2 图元

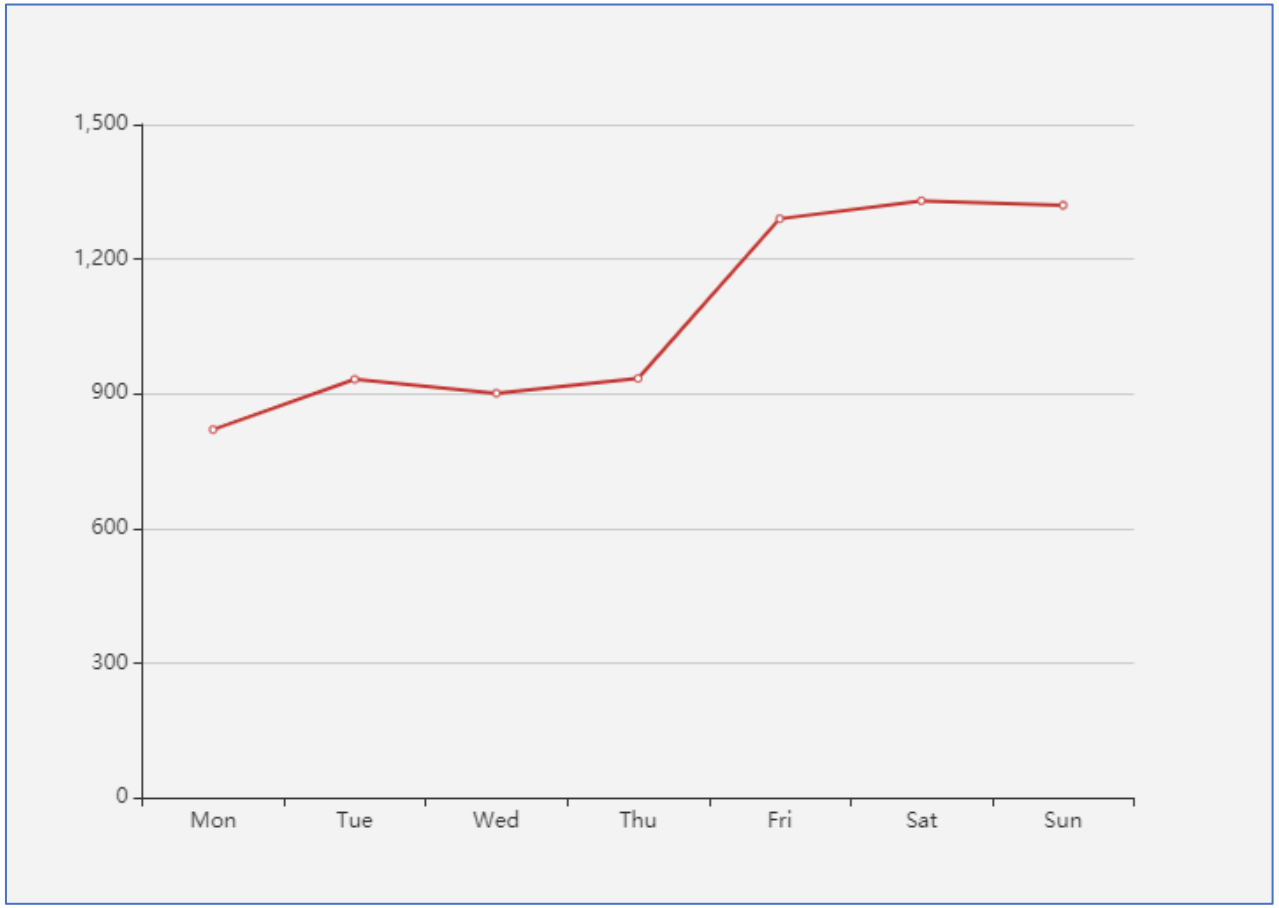
在这些部件中，将业务数据表现为各种形态的基本图形被称为图元(Element)。

比如右图中的圆点：



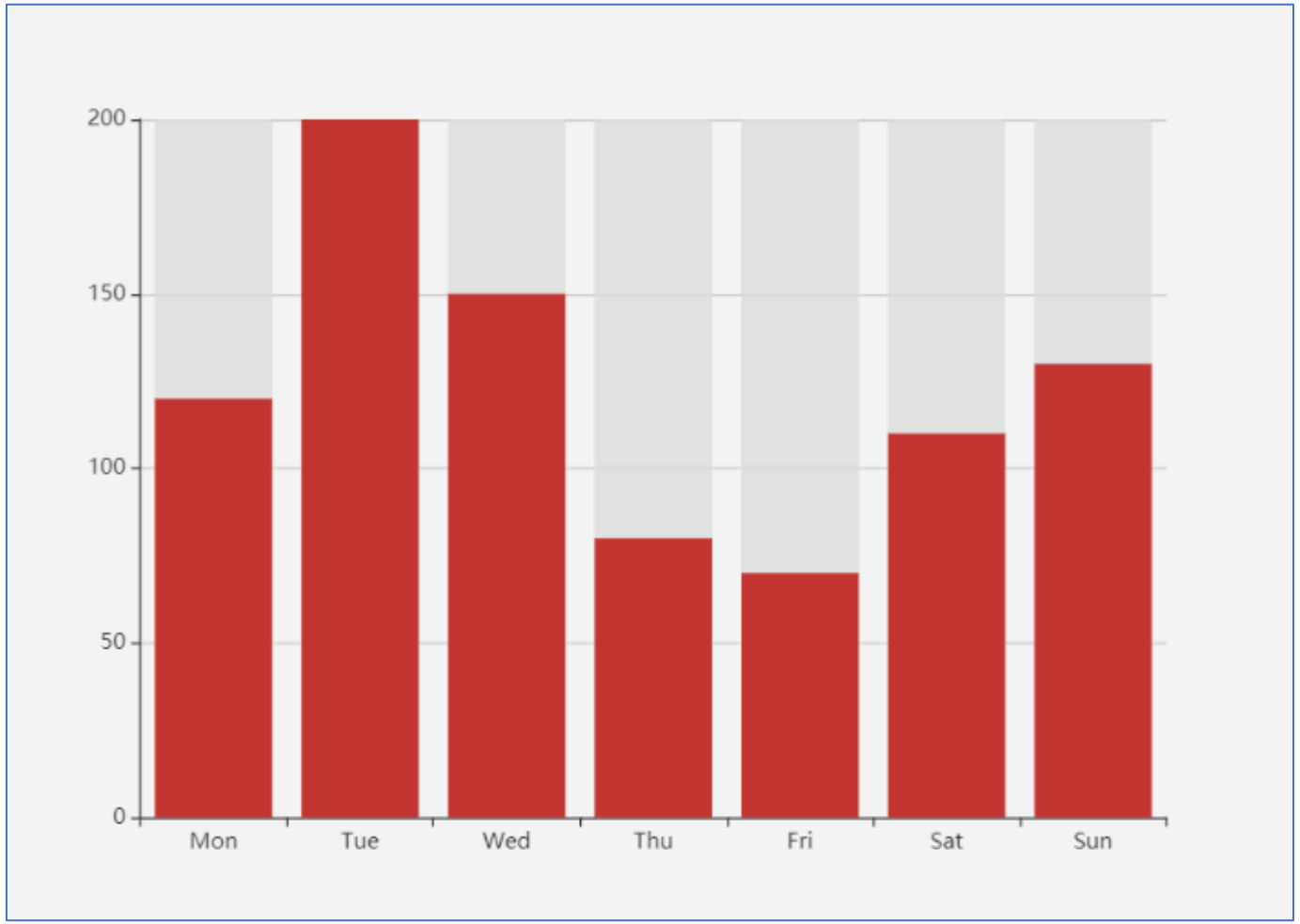
15.1.2 图元

比如右图中的折线：



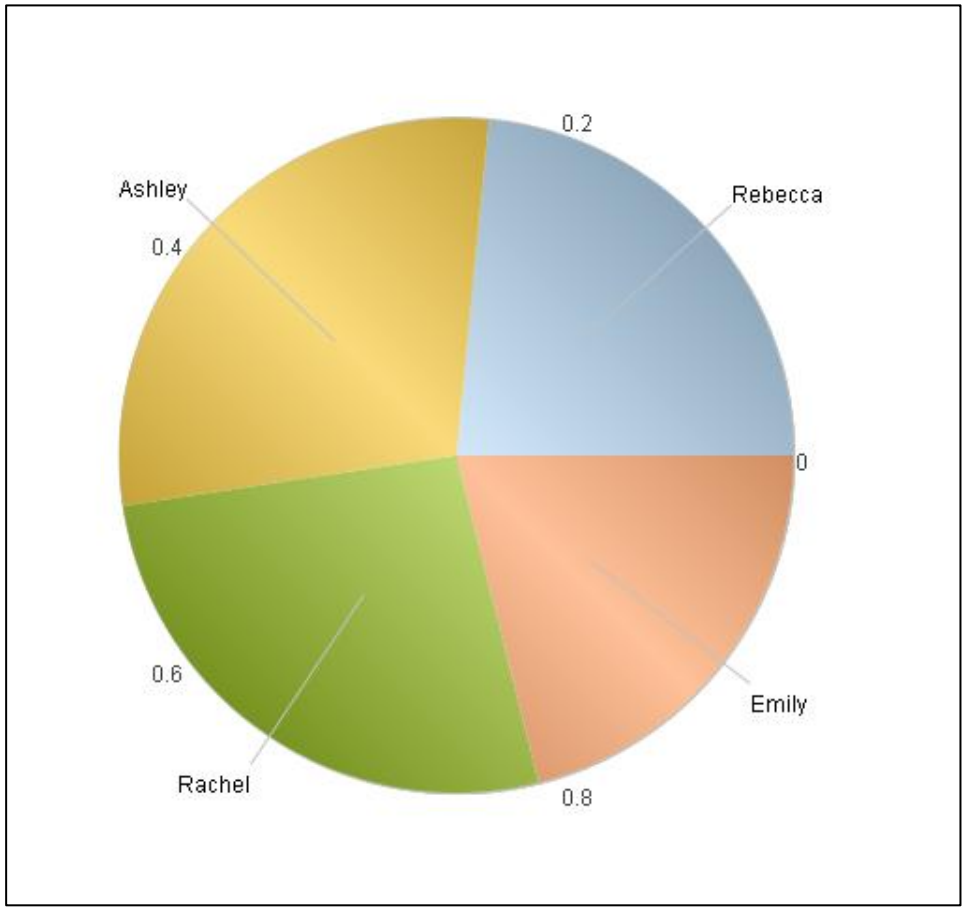
15.1.2 图元

比如右图中的柱形：



15.1.2 图元

比如右图中的饼图：

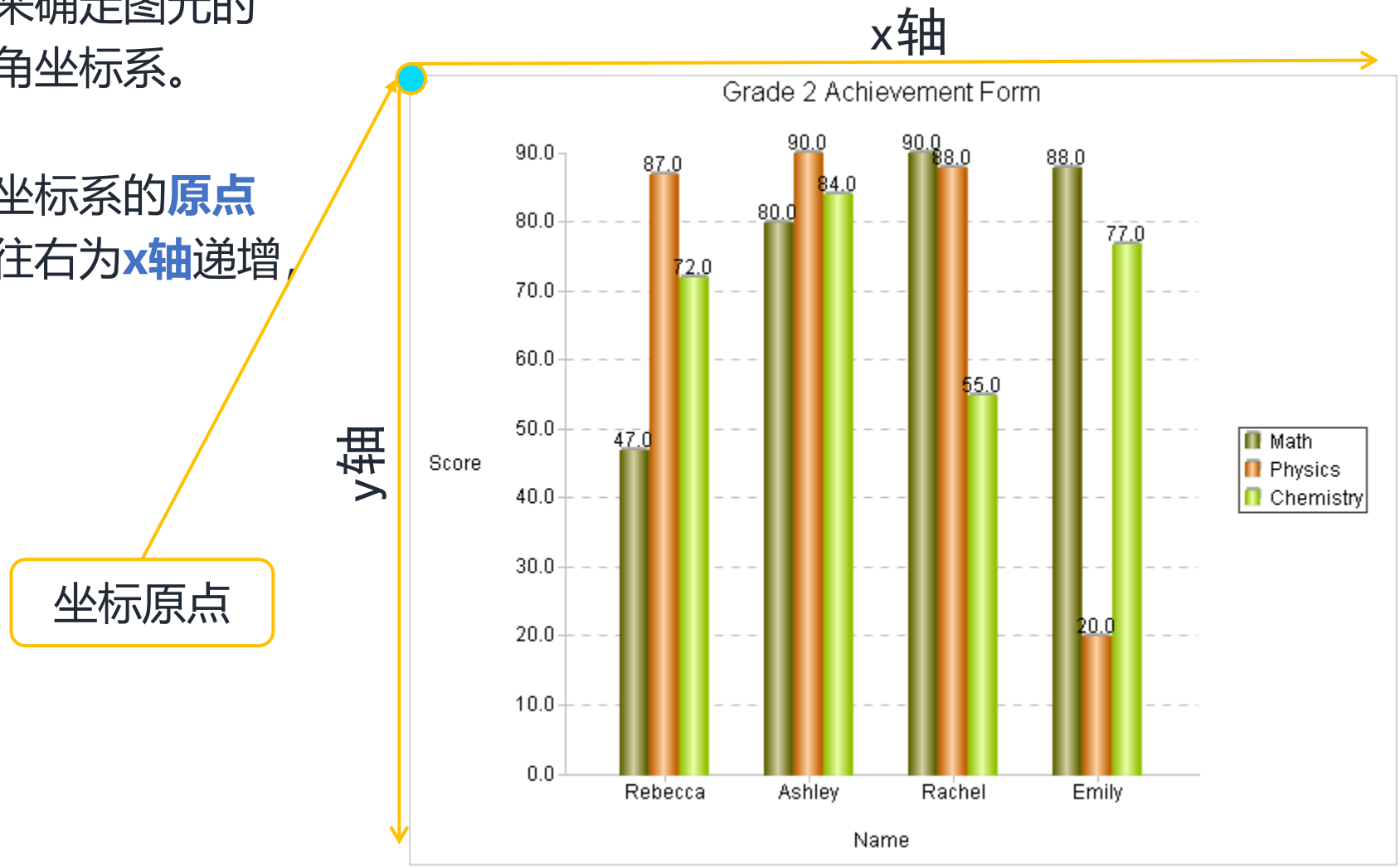


15.2 物理坐标

15.2 物理坐标

在画布上使用**物理坐标系**来确定图元的位置，物理坐标系是个直角坐标系。

不同于常规坐标系，物理坐标系的**原点**通常位于画布的左上角，往右为**x轴**递增，往下为**y轴**递增。



物理坐标的单位是**像素**。

15.2 物理坐标

通过一段代码来认识一下坐标。

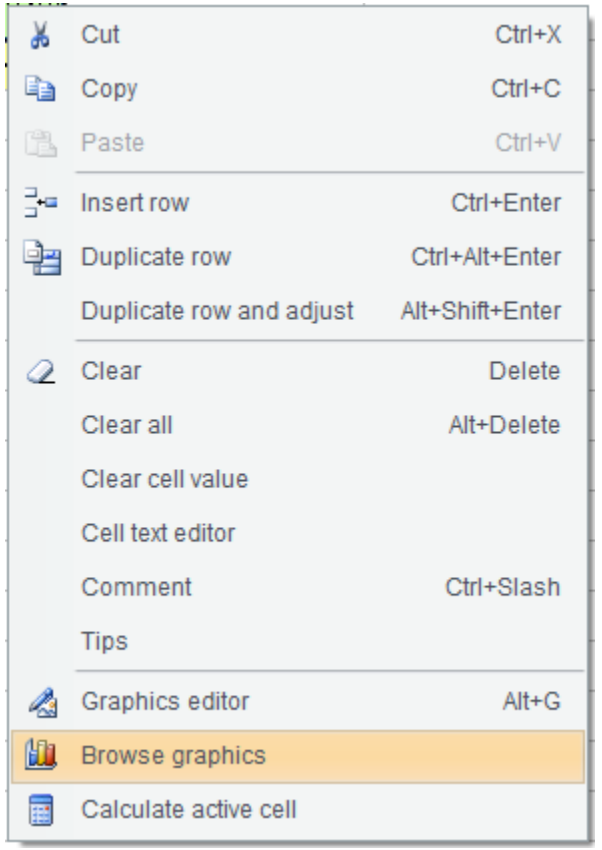
【例15.1】构建一张宽600，高480像素的画布，然后在坐标位置(100,50)处绘制文本Hello, chart!

	A	B
1	=canvas()	/1.创建画布
2	=A1.plot("Text","text":"Hello, chart!","x":100,"y":50)	/2.往画布中布置Text图元并设置坐标和值
3	=A1.draw(600,480)	/3. 画布执行绘制

从示例代码看画图的基本流程：

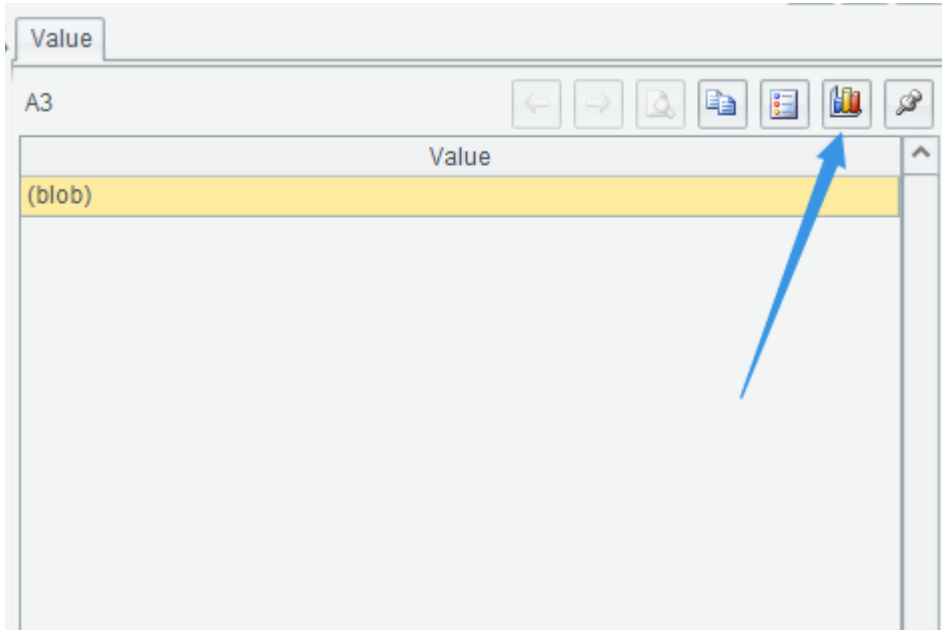
1. 用canvas()创建画布。
2. 往画布里面绘制(plot)文本(Text)，设置相应的参数。
3. 用draw(w,h)绘制出图形。

执行代码后，选中A3格，然后单击快捷菜单项Browse graphics：



15.2 物理坐标

也可以选中A3格，然后单击右边Value面板中的图示图标：



查看A3中的效果图文字示例：

15.2 物理坐标

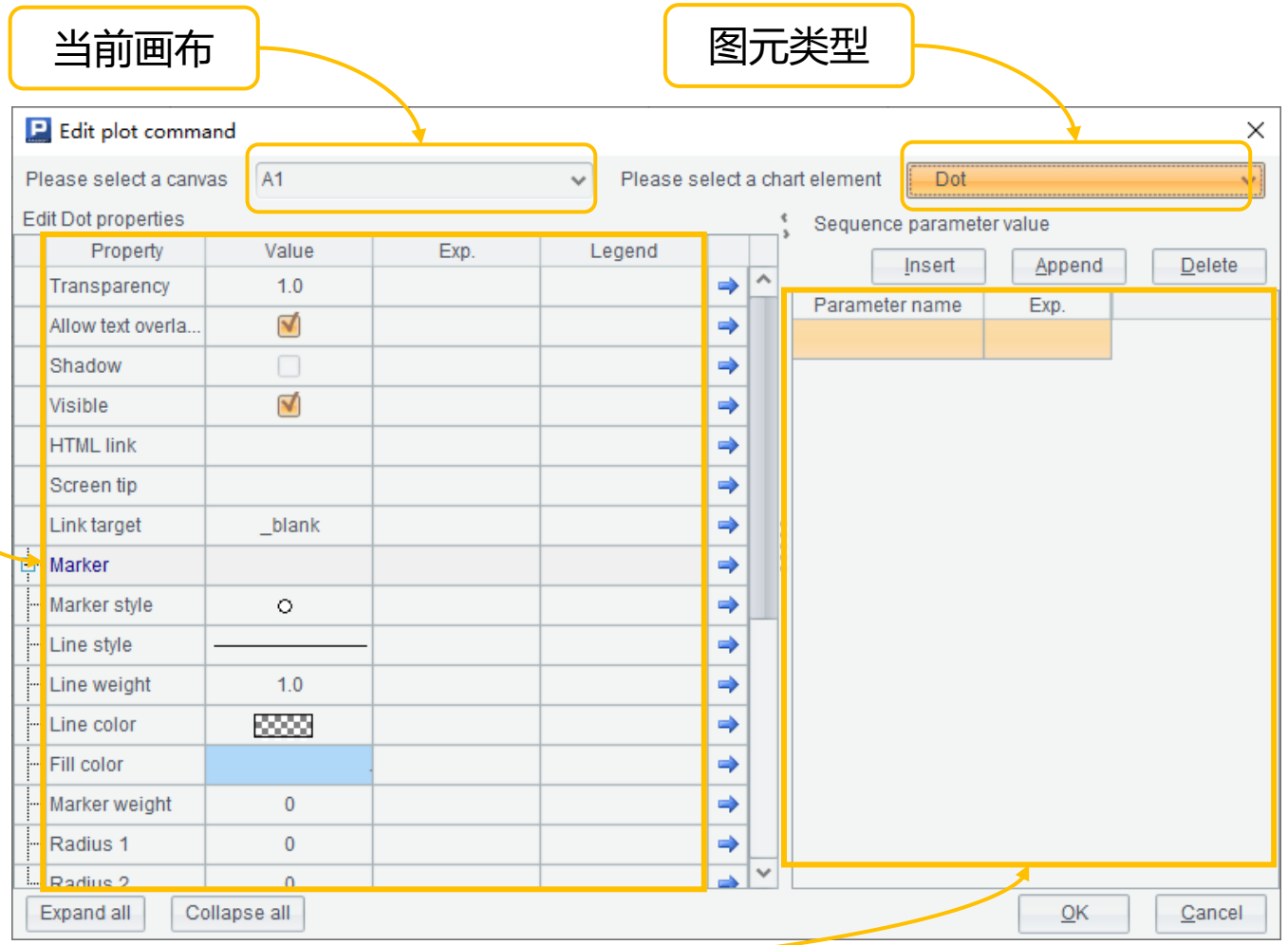
图元属性很多，直接编辑不方便，通常用界面来辅助。

在单元格中用鼠标右键弹出菜单，使用**绘图编辑(Graphics editor)**窗口：

属性编辑区域：
值为常数时，只要编辑Value列。
值需要计算得出时，则要编辑Exp列。

当Value的内容为图形、颜色或者形状时，编辑后会在Exp列显示出当前的表达式描述值。

序列编辑区域：
属性值为序列时，可以点击属性右边的蓝色箭头，然后在这个区域维护序列的成员。



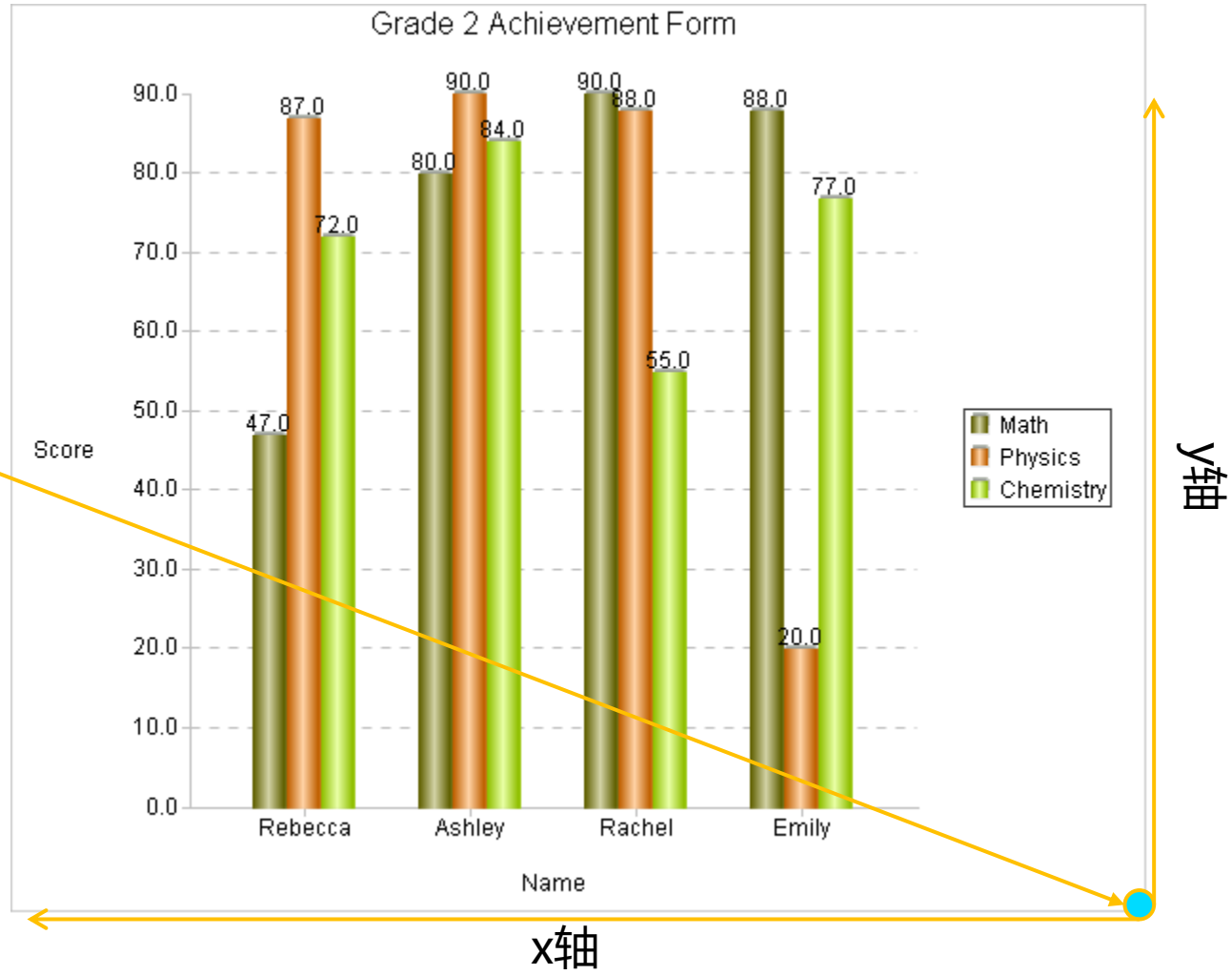
15.2 物理坐标

物理坐标使用负数时表示原点位于画布的右下角，此时往左为x轴绝对值递增，往上为y轴绝对值递增。

注意：

- 坐标值根据正负确定使用的哪个原点。
- 两个坐标值分别有正负时，则是分别使用各自的原点。

坐标值为负数时原点



15.2.1 绝对坐标

物理坐标系中，我们直接采用**像素**作为位置和尺寸的单位，这种表示法称为**绝对坐标**。

【例15.2】在绝对坐标(100,100)处，将文字Hello，chart！绘制为宽高均为100的二维码，并比较不同尺寸画布时的效果。

如图中红框所示，设置文本(Text)图元的属性值。

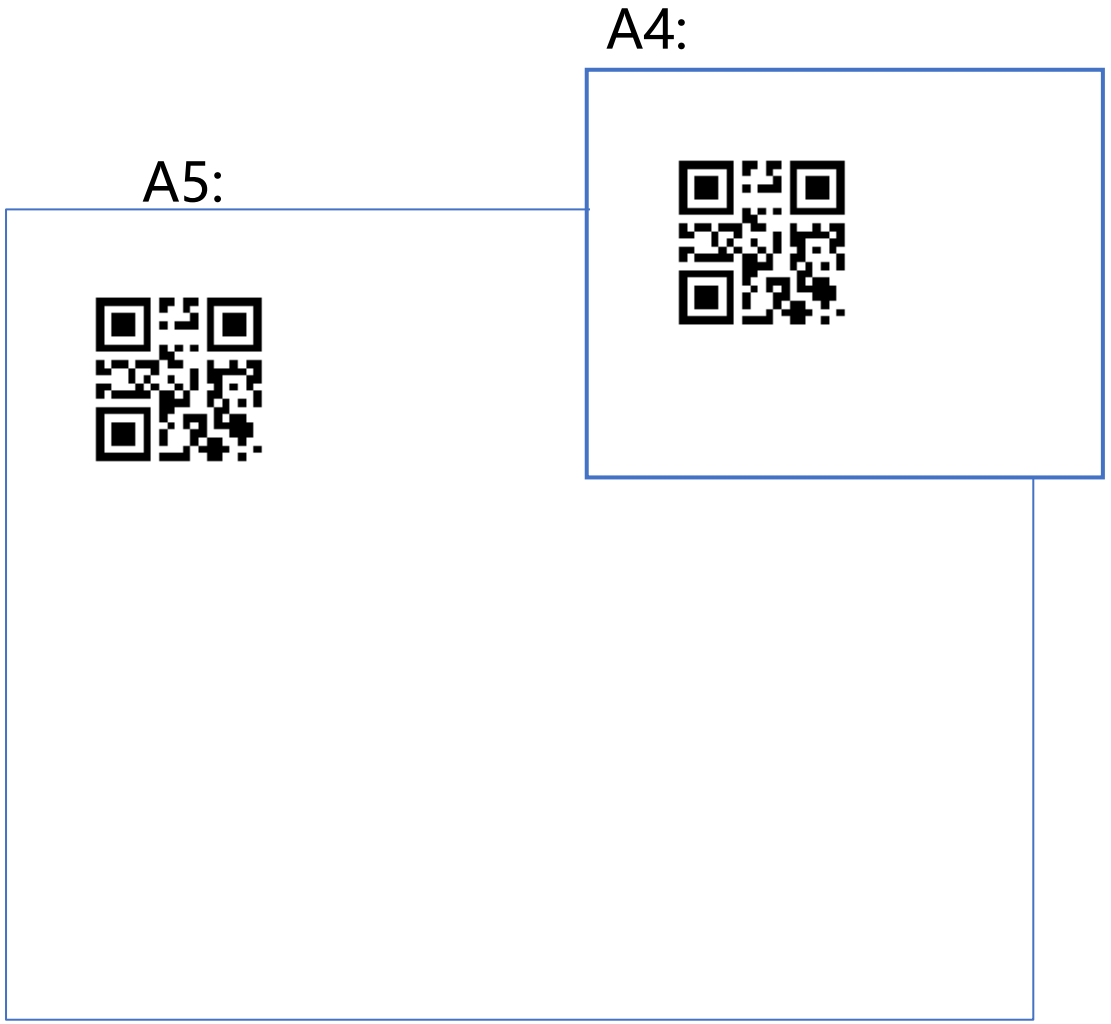
15.2.1 绝对坐标

编辑好的示例脚本：

	A	B
1	=canvas()	/1.创建画布
2	=A1.plot("BackGround")	/2.设置缺省白色的背景
3	=A1.plot("Text","text":"Hello,chart!","x":100,"y":100,"width":100,"height":100,"barType":256)	/3.往画布中布置Text图形元素并设置坐标和值
4	=A1.draw@p(300,240)	/4. 绘制小图形
5	=A1.draw@p(600,480)	/5. 绘制大图形

15.2.1 绝对坐标

执行代码，查看格子A4和A5中图形，可以看到二维码的位置和大小不变。



特征：

- 绝对坐标不会跟随画布大小的变化而变化。
- 绝对坐标的绝对值总是大于1。

15.2.2 相对坐标

采用绝对坐标时，图形不能跟随画布大小成比例变化。为了解决这个问题，我们引入相对坐标。

采用相对于画布宽高的比值来表示的坐标称为**相对坐标**。

【例15.3】使用相对坐标(0.5, 0.5)，总是在中心点将文字Hello，chart！绘制为宽高均为0.2的二维码，注意比较不同画布下的效果。

如图中红框所示，设置文本元素的属性值：

Edit Text properties

Property	Value	Exp.
Text	Hello,chart!	
Visible	☑	
Multiline style	☐	
⊟ Coordinates		
X	0.5	
Y	0.5	
Width	0.2	
Height	0.2	
⊞ Text		
⊞ Alignment		
⊟ Barcode		
Barcode type	QRCode	
Character set	UTF-8	
Show text	☐	

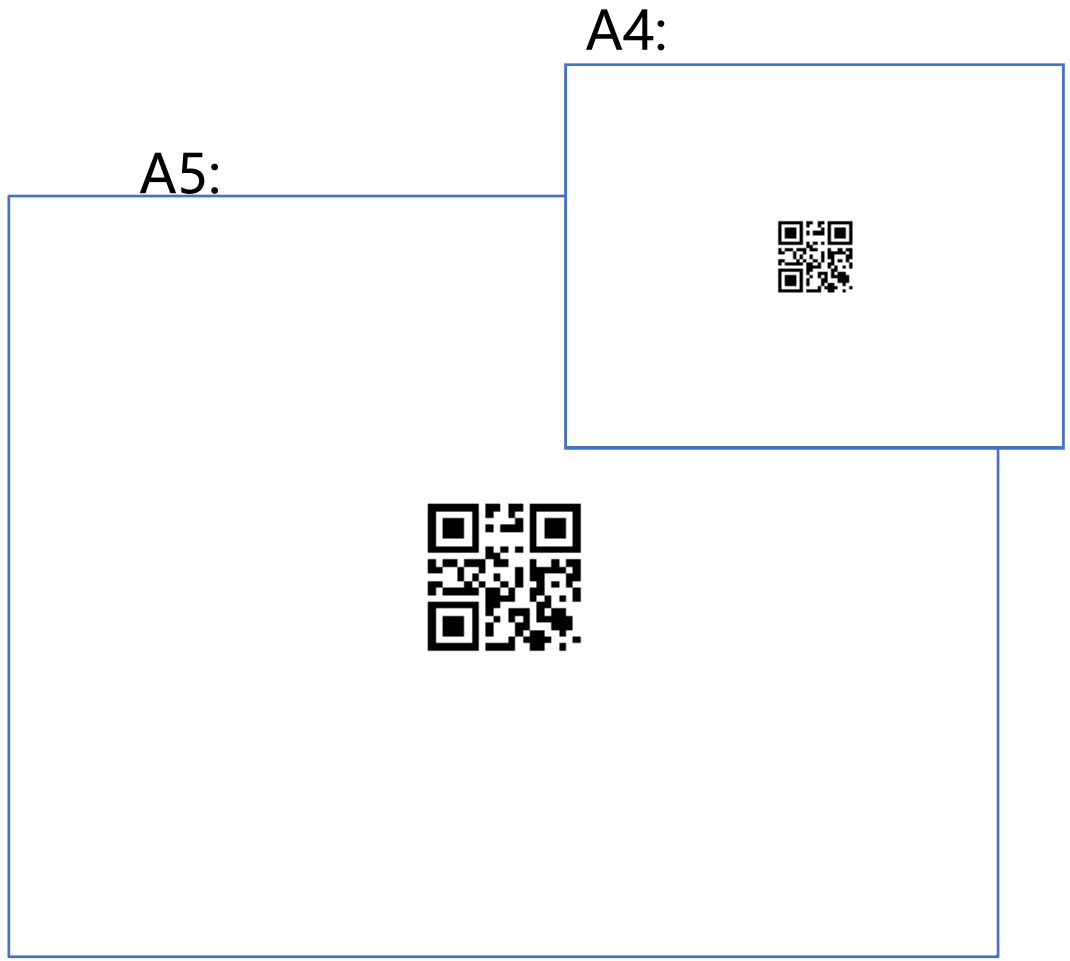
15.2.2 相对坐标

编辑好的示例脚本：

	A	B
1	=canvas()	/1.创建画布
2	=A1.plot("BackGround")	/2.设置缺省白色的背景
3	=A1.plot("Text","text":"Hello,chart!","width":0.2,"height":0.2,"barType":256)	/3.往画布中布置Text图元并设置坐标和值
4	=A1.draw@p(300,240)	/4. 绘制小图形
5	=A1.draw@p(600,480)	/5. 绘制大图形

15.2.2 相对坐标

执行代码，查看格子A4和A5中图形，可以看到二维码的位置和大小跟随画布尺寸成比例变化。



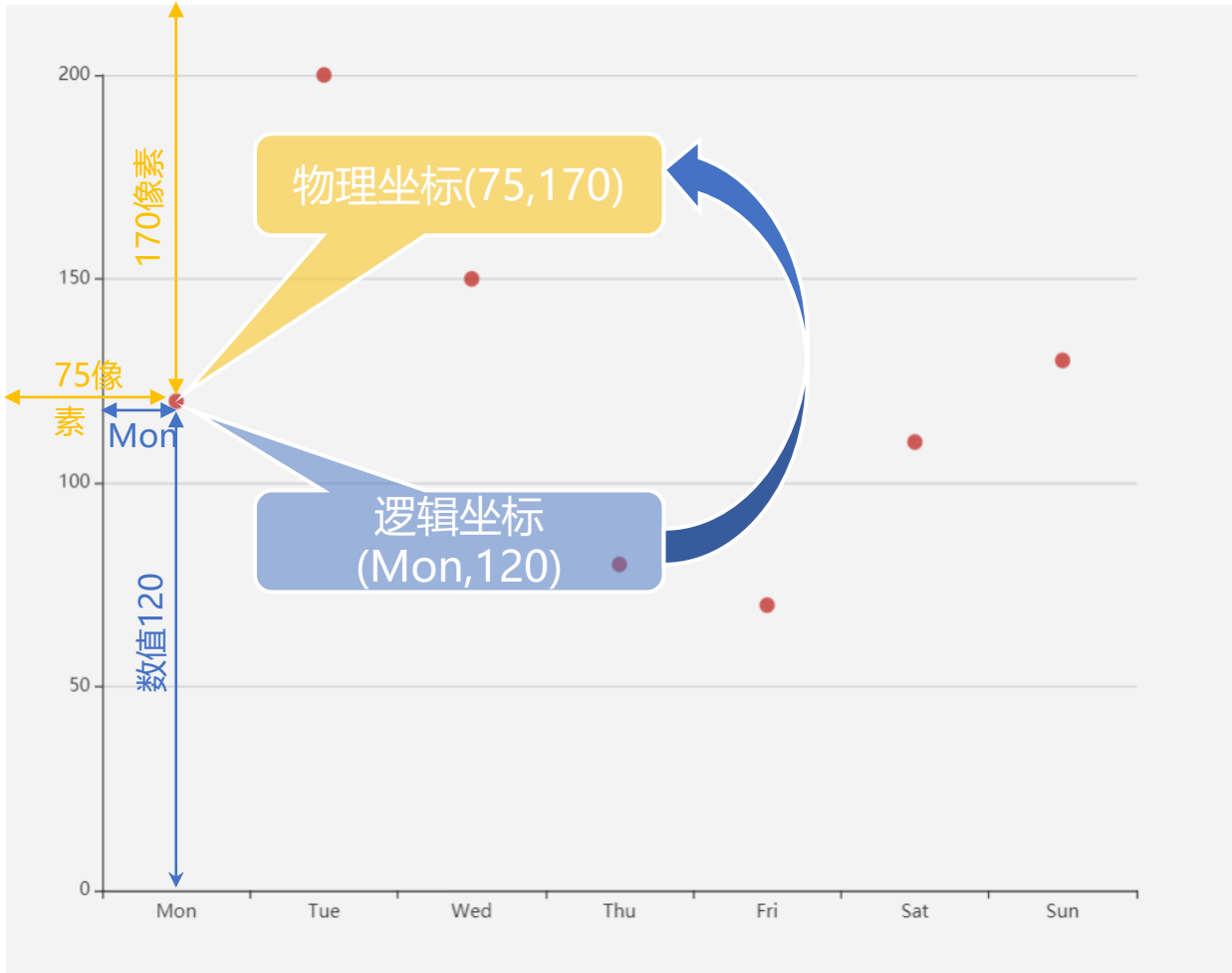
特征：

- 相对坐标会跟随画布的大小成比例变化。
- 相对坐标的绝对值总是小于等于1。

15.3 逻辑坐标

15.3 逻辑坐标

看如下统计图：



第一个点表示业务含义为周一的销量为120。

也就是说，我们希望表示“周一”的横坐标位置和表示120的纵坐标位置上画一个点。

但是，计算机并不认识(周一,120）这样的坐标值，必须要转换成物理坐标，结果在像素值为(75，170)的位置画一个点。

显然，如果有一套坐标系统能让我们直接在(周一,120)的位置上绘制，就会方便很多。

这就要引入逻辑坐标系和逻辑轴的概念。

15.3 逻辑坐标

统计图中能将业务数据（逻辑坐标）转换为物理坐标的坐标系称为**逻辑坐标系。**

构成逻辑坐标系的轴则称为**逻辑轴**。

逻辑轴上对应的坐标则为**逻辑坐标。**

使用逻辑坐标系之后，我们可以直接使用逻辑坐标值来绘图，而由逻辑轴负责将逻辑坐标转换成物理坐标。

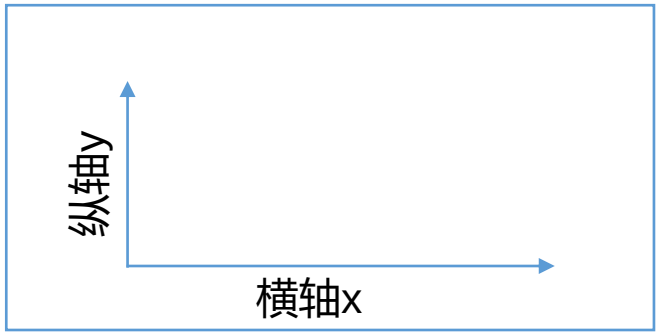
15.3 逻辑坐标

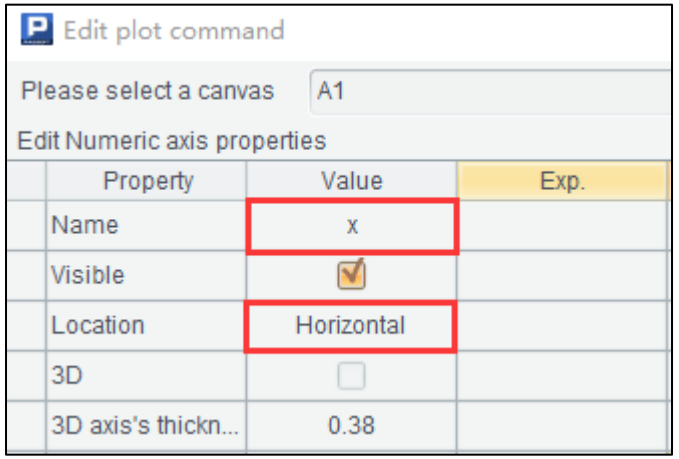
逻辑坐标系仍然是直角坐标，需要一个横逻辑轴和一个纵逻辑轴来确定位置。

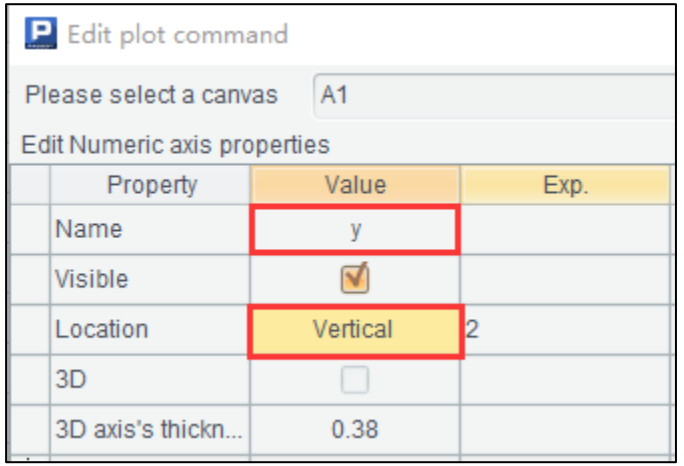
而逻辑轴则由物理坐标来确定其位置。

逻辑轴的缺省物理坐标已经采用相对坐标，设置在画布中的恰当方位。使用缺省坐标时，只要选择轴的位置即可。

如右图所示，设置好两个逻辑轴的名称和位置，便可构成如下图逻辑坐标系：







15.3.1 数值轴

坐标值为数值类型的逻辑轴称为数值轴(Numeric axis)，这是最简单的逻辑轴。它的取值是连续的。

【例15.4】构造直角坐标系，指定横轴的数值范围为0到100，纵轴的范围为0到80，然后在数值坐标(60,64)处绘制一个点。

选择数值轴，使用缺省物理坐标，如右图数值轴的相关属性：

横轴名称设为 x

纵轴名称设为 y

Edit Numeric axis properties

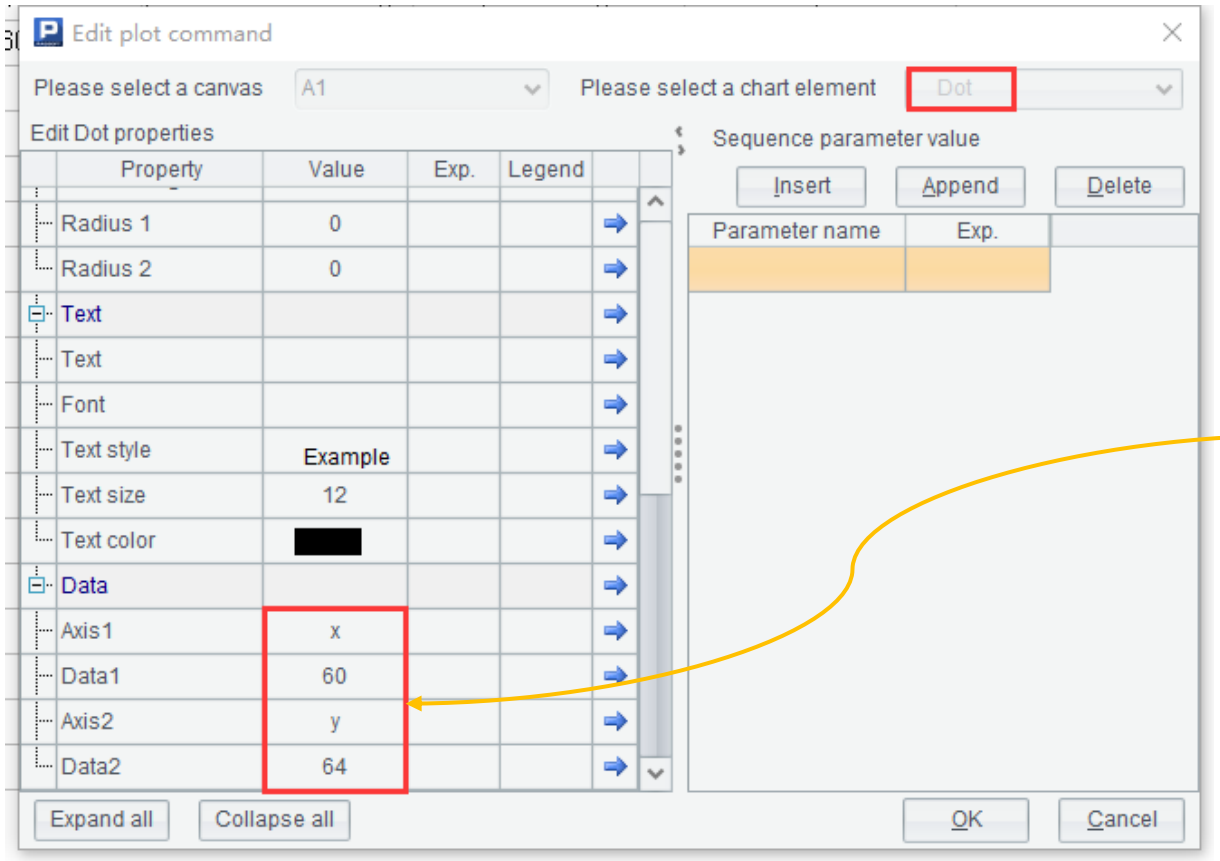
Property	Value	Exp.
Name	x	
Visible	☑	
Location	Horizontal	
3D	☐	
3D axis's thickness ratio	0.38	
Numeric axis		
Autocalc value range	☐	
AutoRange starts from 0	☐	
Max value	100	
Min value	0.0	
Scale count	5	
Format	#.##	

Edit Numeric axis properties

Property	Value	Exp.
Name	y	
Visible	☑	
Location	Vertical	
3D	☐	
3D axis's thickness ratio	0.38	
Numeric axis		
Autocalc value range	☐	
AutoRange starts from 0	☐	
Max value	80	
Min value	0.0	
Scale count	5	
Format	#.##	

15.3.1 数值轴

添加点(Dot)图元，并如图设置属性，由于数据是常数值，因此直接填写在Value列：



- 图元通过Data属性下的Axis1和Axis2，指定为轴x和y，便可确定所属坐标系。
- 每个轴上的业务数据则分别对应Data1和Data2。

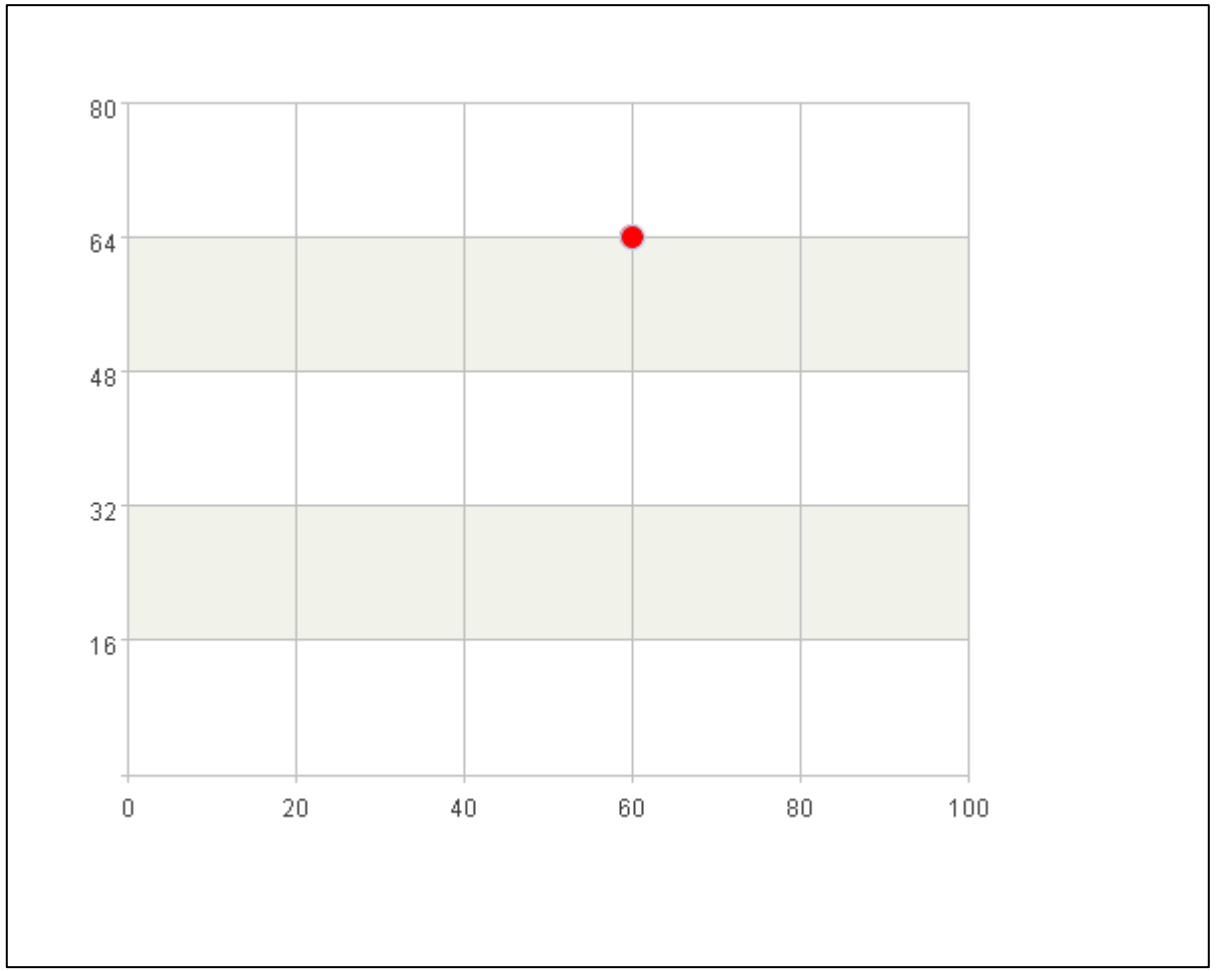
15.3.1 数值轴

编辑完成后，示例脚本如下：

	A	B
1	=canvas()	/创建画布
2	=A1.plot("NumericAxis","name":"x","autoCalcValueRange":false,"autoRangeFromZero":false,"maxValue":100,"xStart":0.1,"labelOverlapping":true)	/布置横轴x
3	=A1.plot("NumericAxis","name":"y","location":2,"autoCalcValueRange":false,"autoRangeFromZero":false,"maxValue":80)	/布置纵轴y
4	=A1.plot("Dot","markerColor":["ChartColor",0,false,-65536,-16777216,0],"markerWeight":6,"axis1":"x","data1":"60","axis2":"y","data2":"64")	/布置点图元并设置数值和坐标系
5	=A1.plot("BackGround")	/缺省白色背景
6	=A1.draw@p(600,480)	/绘制图形

15.3.1 数值轴

执行后查看A6中的**点图**：



15.3.1 数值轴

【示例15.5】构造直角坐标系，随机画3组*20个点。每个点的纵坐标区间都为区间[0,1000]，第1组点的横坐标区间为[0,10]，第2组[0,100]，第3组[0,1000]。

跟上一节类似，只要循环在各区间添加20个随机点即可，示例脚本如下：

	A	B
1	=canvas()	/创建画布
2	=A1.plot("NumericAxis","name":"x","autoCalcValueRange":false,"maxValue":1000,"scaleNum":3)	/布置数值轴x
3	=A1.plot("NumericAxis","name":"y","location":2)	/布置数值轴y
4	for 20	=A1.plot("Dot","axis1":"x","data1":rand(10),"axis2":"y","data2":rand(1000))
5		=A1.plot("Dot","axis1":"x","data1":rand(100),"axis2":"y","data2":rand(1000))
6		=A1.plot("Dot","axis1":"x","data1":rand(1000),"axis2":"y","data2":rand(1000))
7	=A1.plot("BackGround")	
8	=A1.draw@p(600,480)	

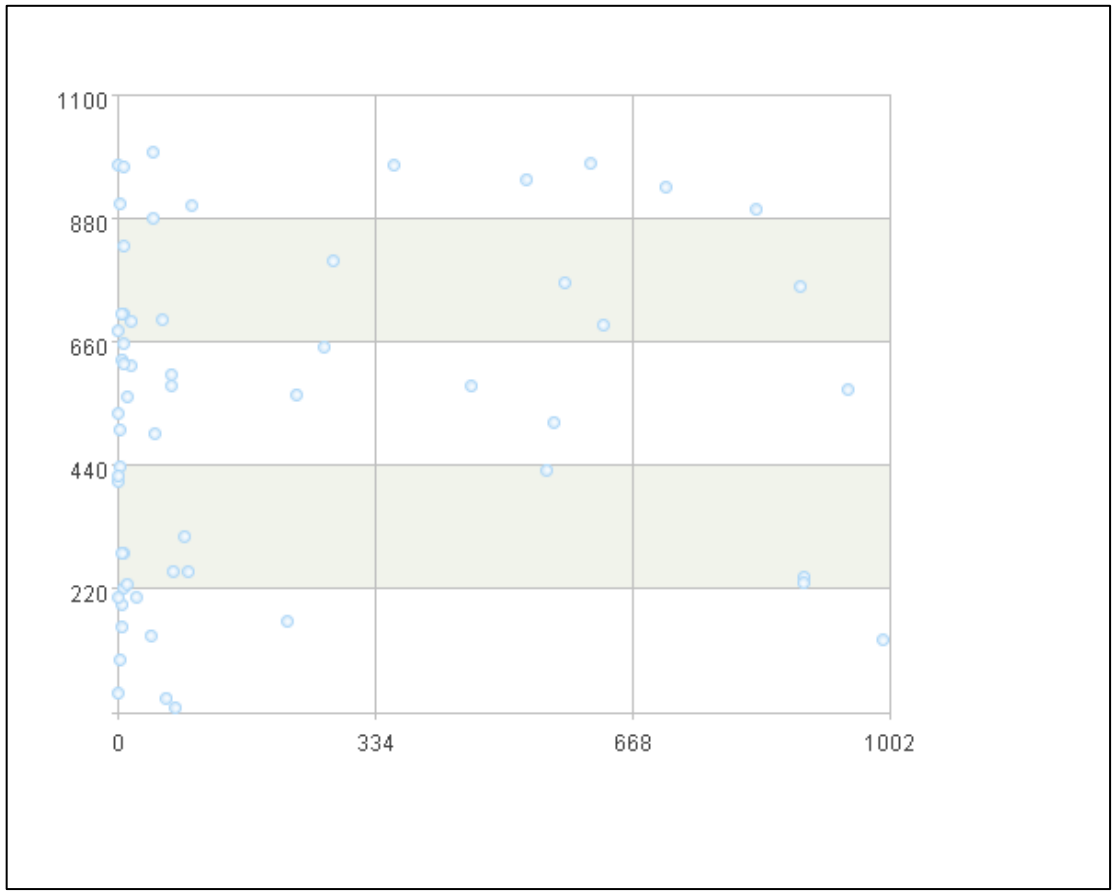
点图元属性设置如图，此时的值为表达式，需要计算，所以填写在Exp列：

Edit Dot properties

Property	Value	Exp.
Transparency	1.0	
Allow text overla...	☑	
Shadow	☐	
Visible	☑	
HTML link		
Screen tip		
Link target	_blank	
⊞ Marker		
⊞ Text		
⊟ Data		
Axis1	x	
Data1		rand(1000)
Axis2	y	
Data2		rand(1000)

15.3.1 数值轴

执行后查看A8中的**随机点图**：



注意：

- 数值轴的坐标范围缺省为根据给定的数值范围自动算出。
- 数值轴的坐标范围也可手动指定。
- 数值轴的值也支持字符串类型的数字。

15.3.1 数值轴

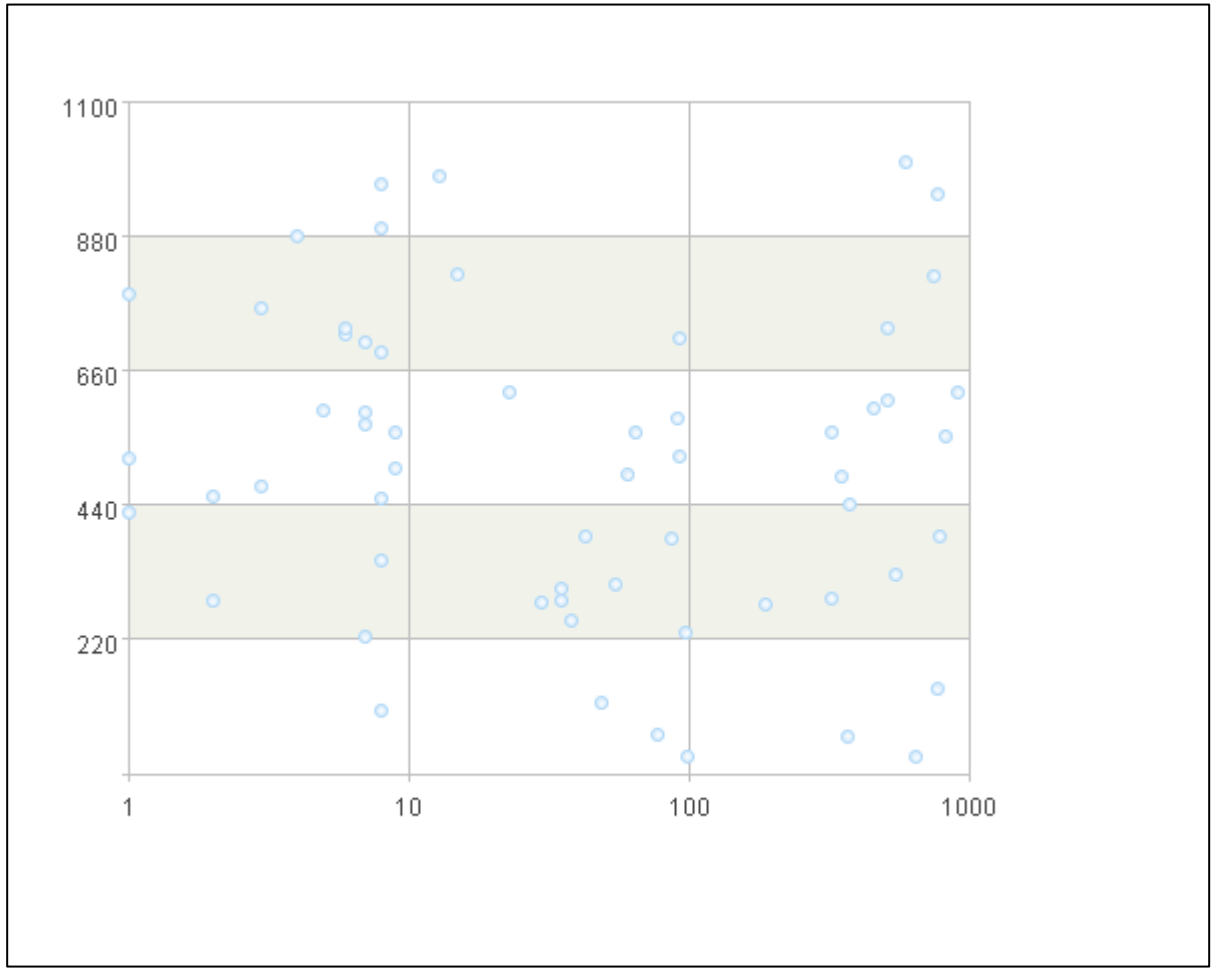
数值轴还可以有非线性变换。

比如上图中的点在x方向的分布不均衡，这时设置坐标轴的变换类型(Transform)，修改上例中x轴的变换类型为Logarithm

Edit Numeric axis properties

Property	Value
Name	x
Visible	☑
Location	Horizontal
3D	☐
3D axis's thickn...	0.38
Numeric axis	
Transform	
Transform	Logarithm
Scale	1.0
Logarithmic ba...	10.0
Power exponent	2.718281828459045

再次执行后查看A8中的按对数变换后均匀分布的随机点图：



15.3.1 数值轴

通常用于绘图的数据会有多个，用循环语句绘制较为复杂，我们还可以使用序列参数，让图元一次性绘制多个点。

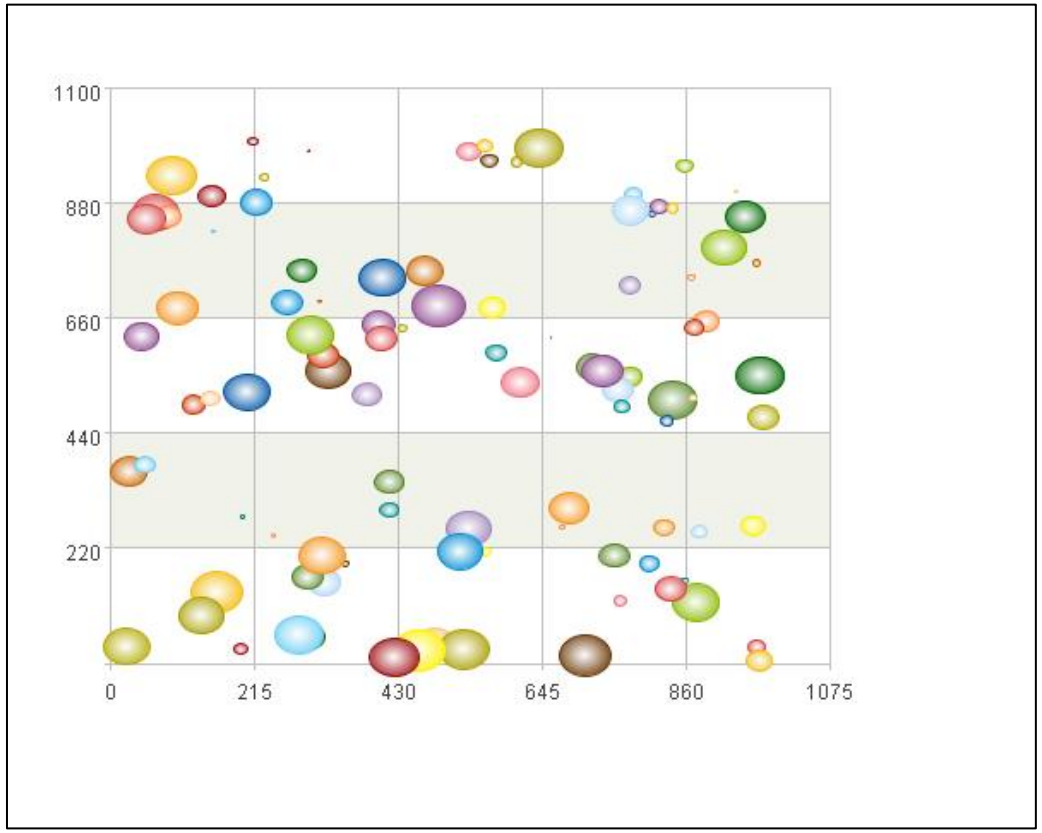
【例15.6】构造直角坐标系，随机绘制100个半径不同的点，使图形展现为**气泡图**。

示例脚本如下：

	A	B
1	>xx=100.(rand(1000))	/产生100个随机范围在1000以内的x坐标
2	>yy=100.(rand(1000))	/y坐标
3	>radius=100.(rand(40))	/点的半径
4	=canvas()	/创建画布
5	=A4.plot("NumericAxis","name":"x")	/布置数值轴x
6	=A4.plot("NumericAxis","name":"y","location":2)	/布置数值轴y
7	=A4.plot("Dot","radius1":radius,"radius2":radius,"axis1":"x","data1":xx,"axis2":"y","data2":yy)	/布置Dot图元，且直接指定数据序列
8	=A4.draw(600,480)	/绘制图形

15.3.1 数值轴

执行后查看A8中的**气泡图**：



注意：

- 点的半径有Radius1和Radius2两个值，分别表示相应轴上的半径值。
- 半径的单位不是像素，同样是业务含义的数值。

15.3.2 枚举轴

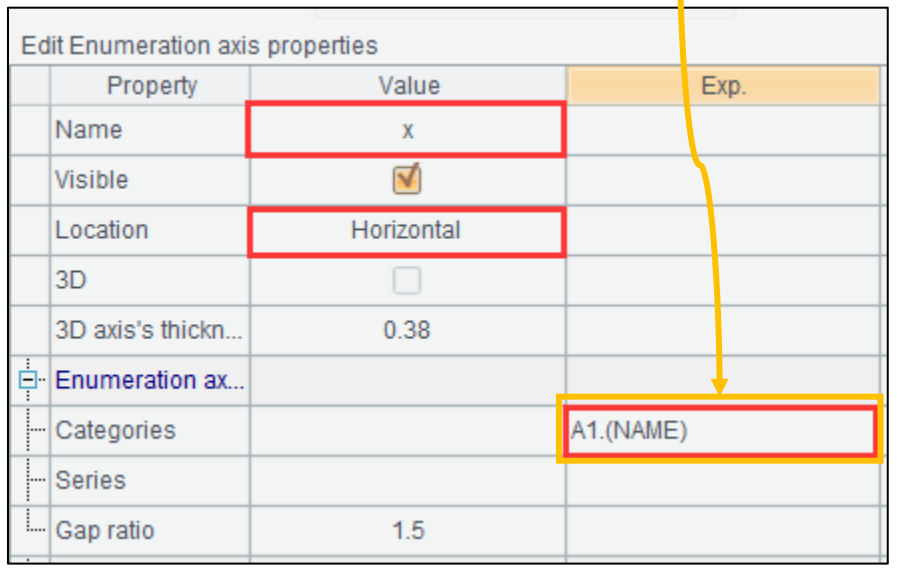
坐标取值只能为序列成员的逻辑轴称为**枚举轴(Enumeration axis)**，枚举轴的取值是不连续的。

枚举轴的坐标值通常又称为**分类**，比如分类Rebecca。

【例15.7】将总分表(summary.txt)的数据绘制成**折线**。数据如右：

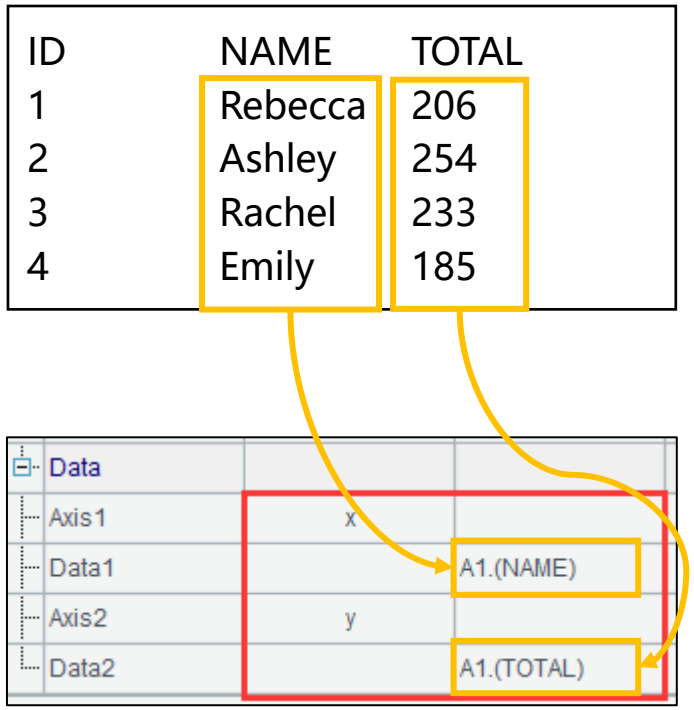
ID	NAME	TOTAL
1	Rebecca	206
2	Ashley	254
3	Rachel	233
4	Emily	185

设置枚举轴的名称，位置和分类数据，以及分类的数据内容示意如右图：



15.3.2 枚举轴

然后编辑Line图元，并设置相关坐标系以及数据表达式，对应坐标轴的数据示意图如下：

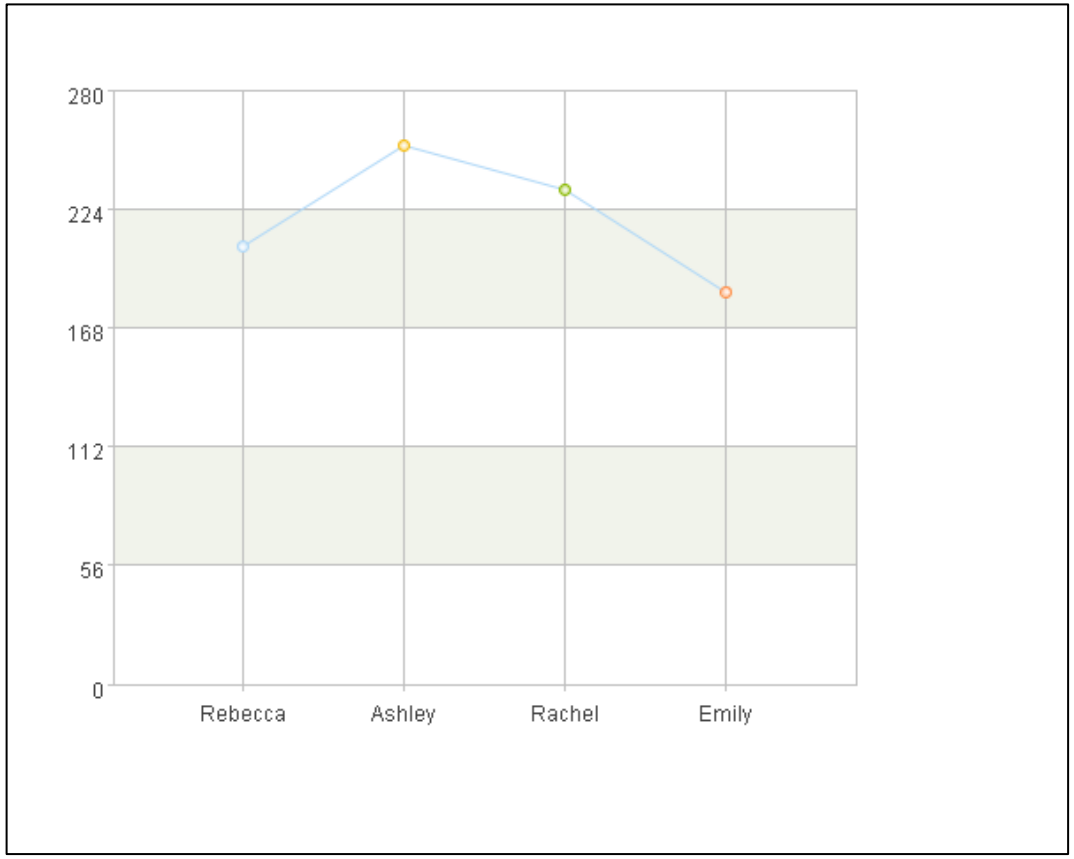


编辑好的示例脚本如下：

	A	B
1	=file("summary.txt").import@t()	/1.导入数据
2	=canvas()	/2.创建画布
3	=A2.plot("EnumAxis","name":"x","categories":A1.(NAME))	/3.枚举型横轴x
4	=A2.plot("NumericAxis","name":"y","location":2)	/4.数值型纵轴y
5	=A2.plot("Line","axis1":"x","data1":A1.(NAME),"axis2":"y","data2":A1.(TOTAL))	/5.在画布中布置Line图元，设置数据且关联坐标系
6	=A2.draw(600,480)	/6. 绘制图形

15.3.2 枚举轴

执行后查看A6中的**折线图：**



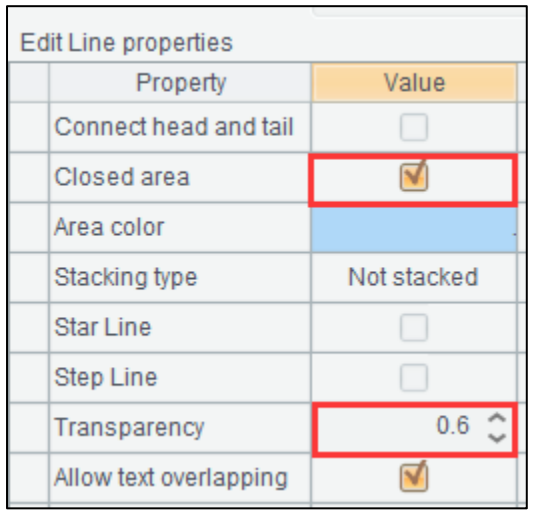
注意：

- 枚举轴的分类和系列内容不指定时，便会根据对应图元的相应数据自动计算出来。
- 如果手动指定枚举轴的分类或系列内容，这个范围必须包含实际图元的数据内容。

15.3.2 枚举轴

【例15.8】将折线图的属性封闭区域(Closed area)勾上，使得折线的展现为面积图。

只需增加【例15.7】 脚本中A5格的两个属性，增加的属性设置如下图：

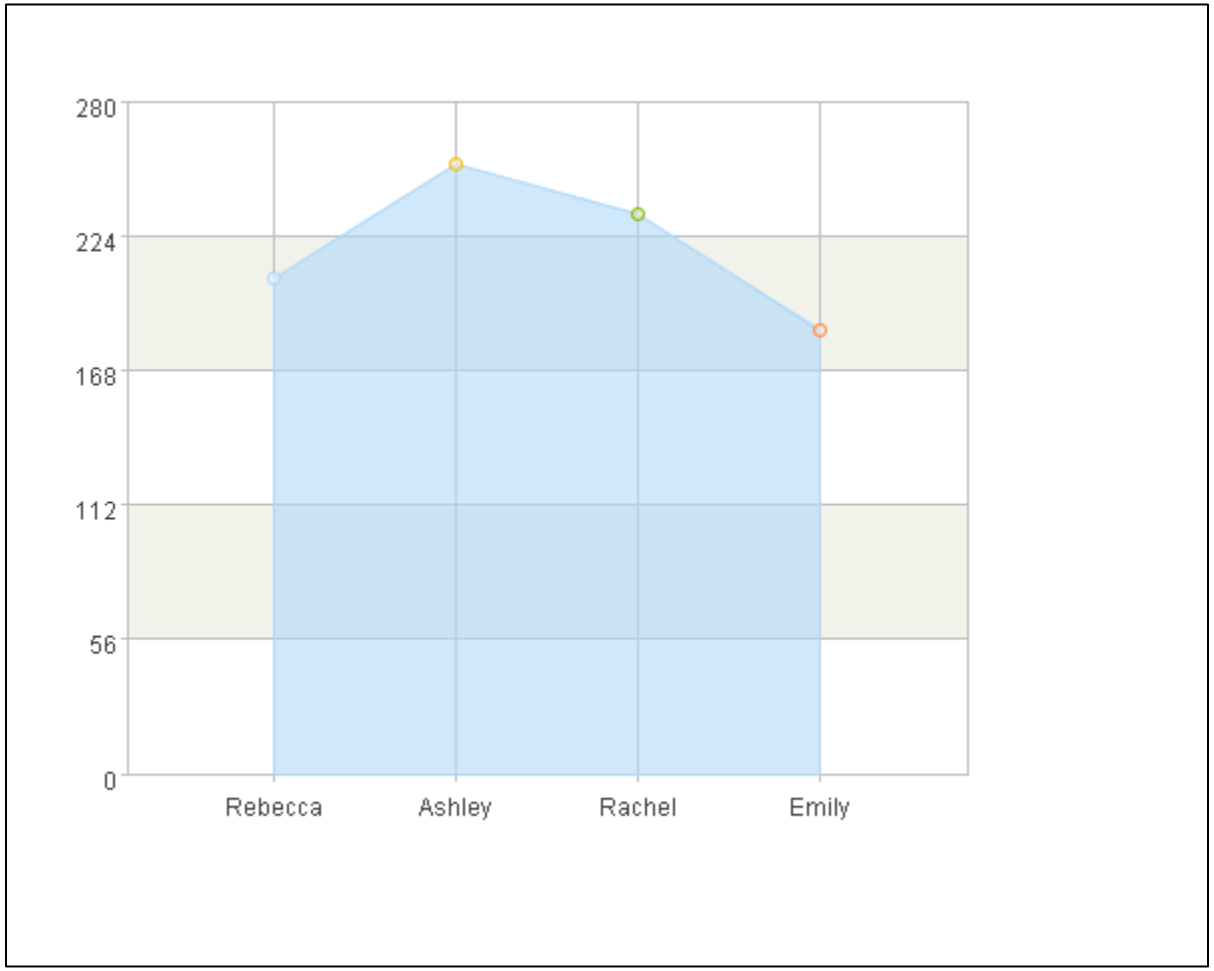


示例脚本如下：

	A	B
1	=file("summary.txt").import@t()	/1.导入数据
2	=canvas()	/2.创建画布
3	=A2.plot("EnumAxis","name":"x")	/3.枚举型横轴x
4	=A2.plot("NumericAxis","name":"y","location":2)	/4.数值型纵轴y
5	=A2.plot("Line","closedArea":true,"transparent":0.6,"axis1":"x","data1":A1.(NAME),"axis2":"y","data2":A1.(TOTAL))	/5.在画布中布置Line图元，设置数据且关联坐标系
6	=A2.draw(600,480)	/6. 绘制图形

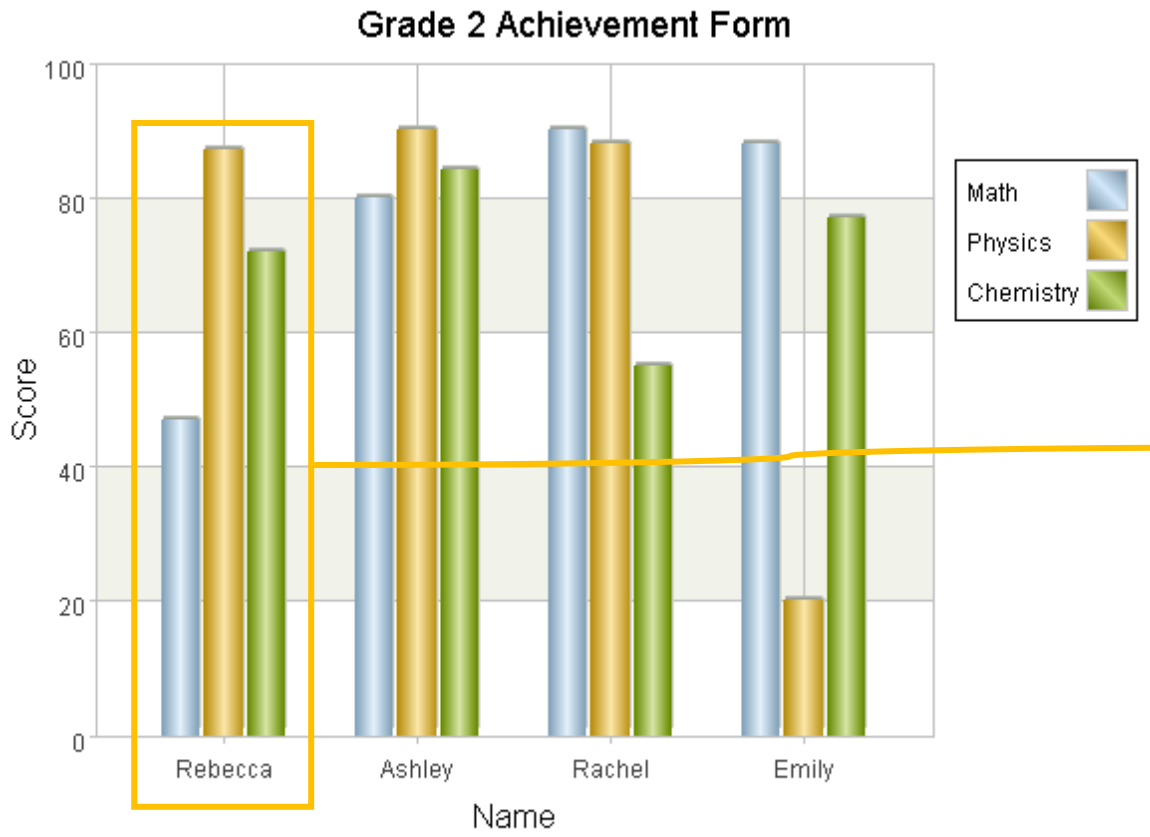
15.3.2 枚举轴

执行后查看A6中的**面积图**：



15.3.2 枚举轴

观察如下统计图：



分类坐标Rebecca下面拥有3根柱子，分别对应Math、Physics和Chemistry。

也就是枚举轴的数据可以有两层，此时第一层称为分类，第二层称为系列。

15.3.2 枚举轴

【例15.9】将分数表(scores.txt)的各科成绩用柱图元绘制成多系列柱图，并实现上一页中完整统计图的效果。

分数表数据如下：

ID	NAME	Math	Physics	Chemistry
1	Rebecca	47	87	72
2	Ashley	80	90	84
3	Rachel	90	88	55
4	Emily	88	20	77

分类值

系列值

如下图设置枚举轴的属性，其中分类和系列的数据也可以省略，省略后，自动从关联图元中获取：

Edit Enumeration axis properties

Property	Value	Exp.
Name	x	
Visible	☑	
Location	Horizontal	
3D	☐	
3D axis's thicknes...	0.38	
Enumeration axis		
Categories		A1.(NAME)
Series		["Math","Physics","Chemistry"]
Gap ratio	1.5	

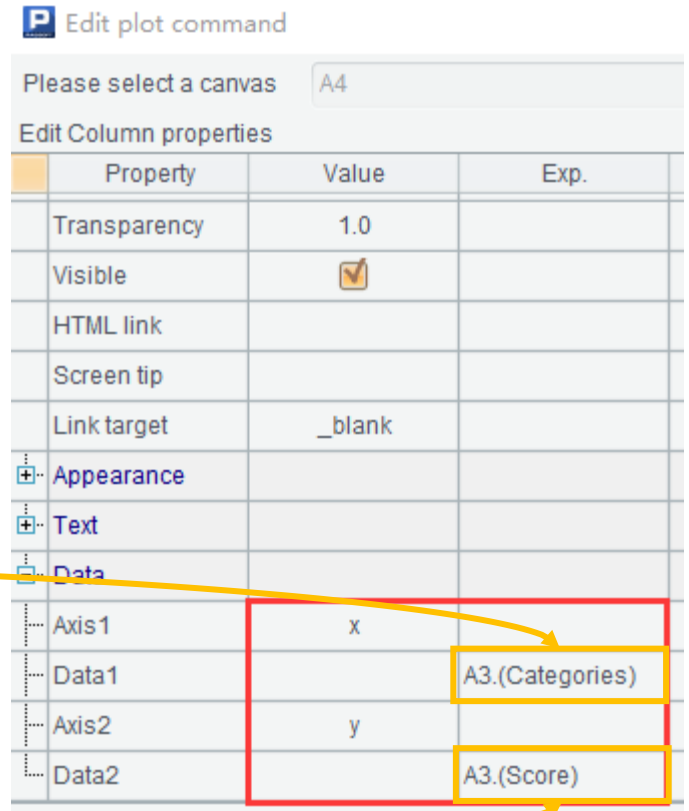
15.3.2 枚举轴

图元中对应枚举轴的数据包含两层时，需要用逗号将分类和系列分开。
比如“Rebecca,Math” 。

然后如下图编辑好Column图元的数据属性，以及对应的数据序列示意图：

绘图前先将数据转换为下图结构：

NAME	Subject	Score	Categories
Rebecca	Math	47	Rebecca,Math
Rebecca	Physics	87	Rebecca,Physics
Rebecca	Chemistry	72	Rebecca,Chemistry
Ashley	Math	80	Ashley,Math
Ashley	Physics	90	Ashley,Physics
Ashley	Chemistry	84	Ashley,Chemistry
Rachel	Math	90	Rachel,Math
Rachel	Physics	88	Rachel,Physics
Rachel	Chemistry	55	Rachel,Chemistry
Emily	Math	88	Emily,Math
Emily	Physics	20	Emily,Physics
Emily	Chemistry	77	Emily,Chemistry



15.3.2 枚举轴

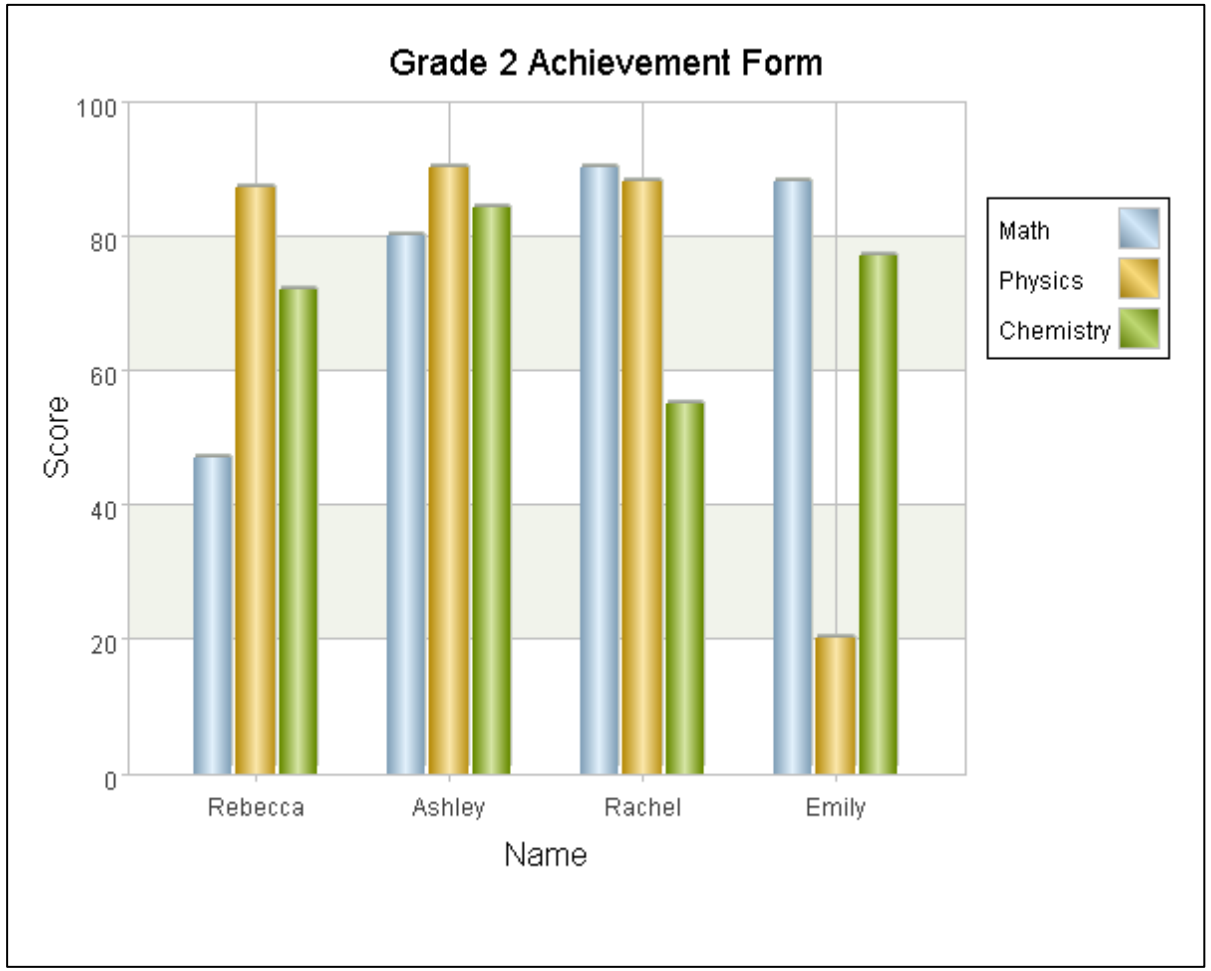
编辑好的示例脚本如下：

	A	B
1	=file("scores.txt").import@t()	/导入分数表
2	=A1.pivot@r(ID,NAME;Subject,Score)	/将分数交叉表转置
3	=A2.derive(NAME+","+Subject:Categories)	/增加Categories字段
4	=canvas()	/创建画布
5	=A4.plot("EnumAxis","name":"x","categories":A1.(NAME),"series":["Math","Physics","Chemistry"],"title":"Name")	/枚举型横轴，命名为x轴，设置轴标题
6	=A4.plot("NumericAxis","name":"y","location":2,"title":"Score","titleAngle":-90)	/数值型纵轴，命名为y轴，设置轴标题以及旋转角度
7	=A4.plot("Column","axis1":"x","data1":A3.(Categories),"axis2":"y","data2":A3.(Score))	/在画布中布置Column图元，设置数据且关联坐标系
8	=A4.plot("Legend")	/布置缺省图例
9	=A4.plot("Text","text":"Grade 2 Achievement Form","y":0.05,"textSize":18)	/用文本绘制统计图标题
10	=A4.plot("BackGround")	/设置缺省白色背景
11	=A4.draw@p(600,480)	/绘制图形

这里用到了图例的缺省绘制，详细用法参见【15.4.3】图例

15.3.2 枚举轴

执行后查看A11中的**多系列柱状图：**



15.3.3 日期轴

坐标取值为日期类型的逻辑轴称为**日期轴(Date axis)**。它的取值跟数值轴类似，也是连续的。

【例15.10】将工程项目进度计划表，用甘特图绘制出来。

数据为CSV格式，如下：

```
Task Description,Start,Duration,End
Brainstorm,7/16/2013,5,7/21/2013
Outline,7/21/2013,10,7/31/2013
First Draft,7/25/2013,7,8/1/2013
First Draft Edit,8/1/2013,3,8/4/2013
Final Draft,8/4/2013,7,8/11/2013
```

- 甘特图的横轴使用日期轴类型。
- 使用**柱(Column)**图元来描述项目的计划，需要用到**Data3**属性表示柱子的开端。
- 从文本导入的数据包含日期时，注意将选项中的**日期格式(Date format)**设置为跟数据一致，否则识别不出日期类型的数据。

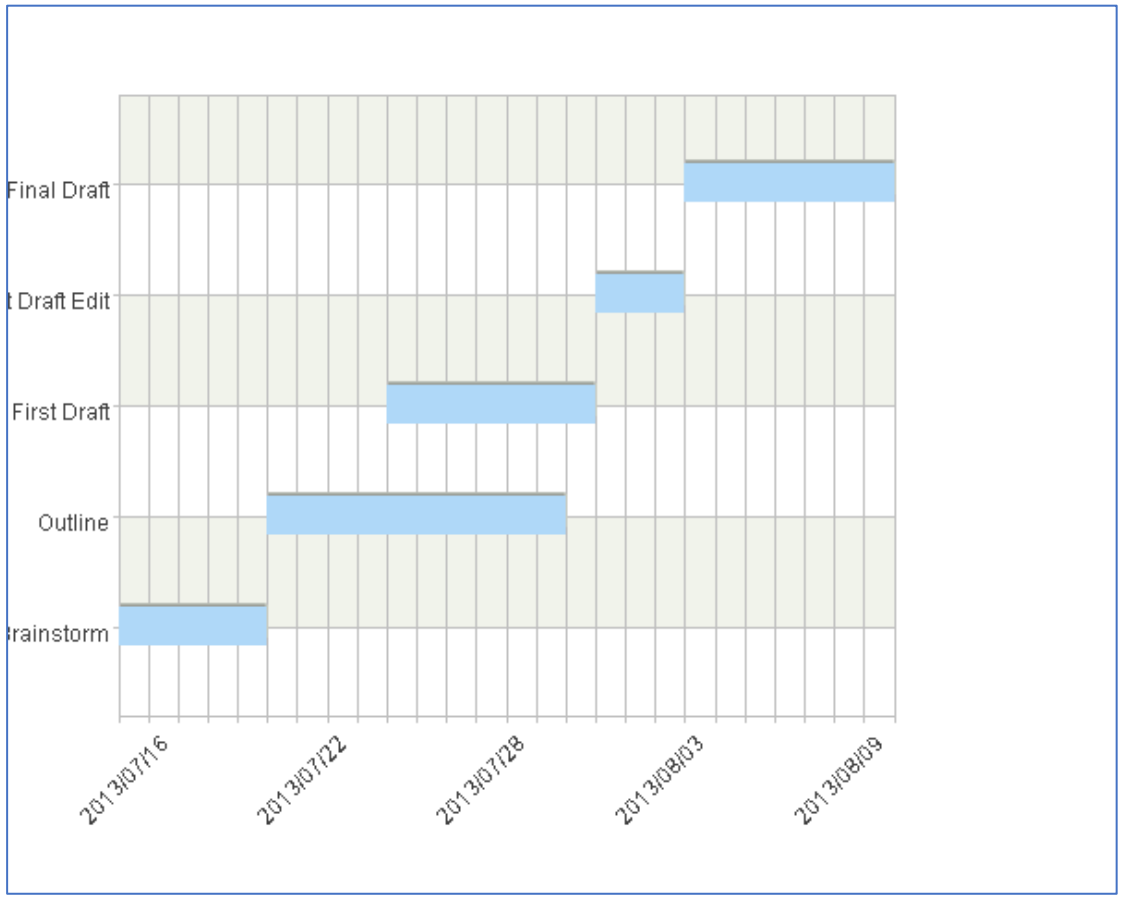
15.3.3 日期轴

示例脚本如下：

	A	B
1	=file("project.csv").import@t(;,",")	/1.导入CSV数据
2	=canvas()	/2.创建画布
3	=A2.plot("DateAxis","name":"x","labelStep":2,"labelAngle":-45)	/3.日期型横轴x
4	=A2.plot("EnumAxis","name":"y","location":2)	/4.枚举型纵轴y
5	=A2.plot("Column","data3":A1.(End),"fillColor":["ChartColor",0,false,-5252872,-5252872,0],"axis1":"y","data1":A1.(#1),"axis2":"x","data2":A1.(#2))	/5.在画布中布置Column图元，设置数据且关联坐标系
6	=A2.plot("Background")	/6.添加白色背景
7	=A2.draw@p(600,480)	/7. 绘制图形

15.3.3 日期轴

执行后查看A7中的**甘特图**：

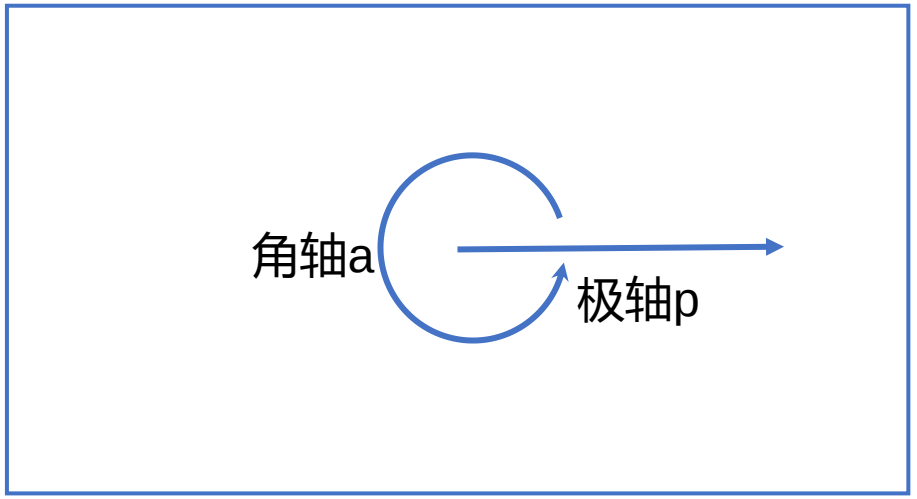


15.4 更多逻辑映射

15.4.1 极坐标系

极坐标系同样由两个**轴**构成。根据轴的**定位方式(Location)**，极坐标系分别由**极轴(Polar)**和**角轴(Angle)** 构成。

如右图设置轴的位置分别为极轴和角轴，便可构成极坐标系:



Edit Enumeration axis properties

Property	Value	Exp.
Name	p	
Visible	☑	
Location	Polar	
3D	☐	
3D axis's thickn...	0.38	

Edit Numeric axis properties

Property	Value	Exp.
Name	a	
Visible	☑	
Location	Angle	
3D	☐	
3D axis's thickn...	0.38	

15.4.1 极坐标系

雷达图其实就是**折线图(Line)**在极坐标系的变形。当设置折线图的枚举轴类型为角轴，数值轴类型为极轴。然后再设置Line图元的**首尾相连(End to end)** 属性，展现的图形即为雷达图。

【例15.11】构造极坐标系，将总分表(summary.txt)的数据绘制成**雷达图**。

各属性设置如图：

折线图元属性：

Edit Line properties

Property	Value	Exp.
End to end	☑	
Closed area	☑	
Area color		
Stacking type	Not stacked	
Star Line	☐	
Step Line	☐	
Transparency	0.6	
Allow text overla...	☑	
Shadow	☐	
Visible	☑	

Edit Enumeration axis properties

Property	Value	Exp.
Name	a	
Visible	☑	
Location	Angle	
3D	☐	

Edit Numeric axis properties

Property	Value	Exp.
Name	p	
Visible	☑	
Location	Polar	
3D	☐	

Edit Line properties

Property	Value	Exp.
Data		
Axis1	a	
Data1		A1.(NAME)
Axis2	p	
Data2		A1.(TOTAL)

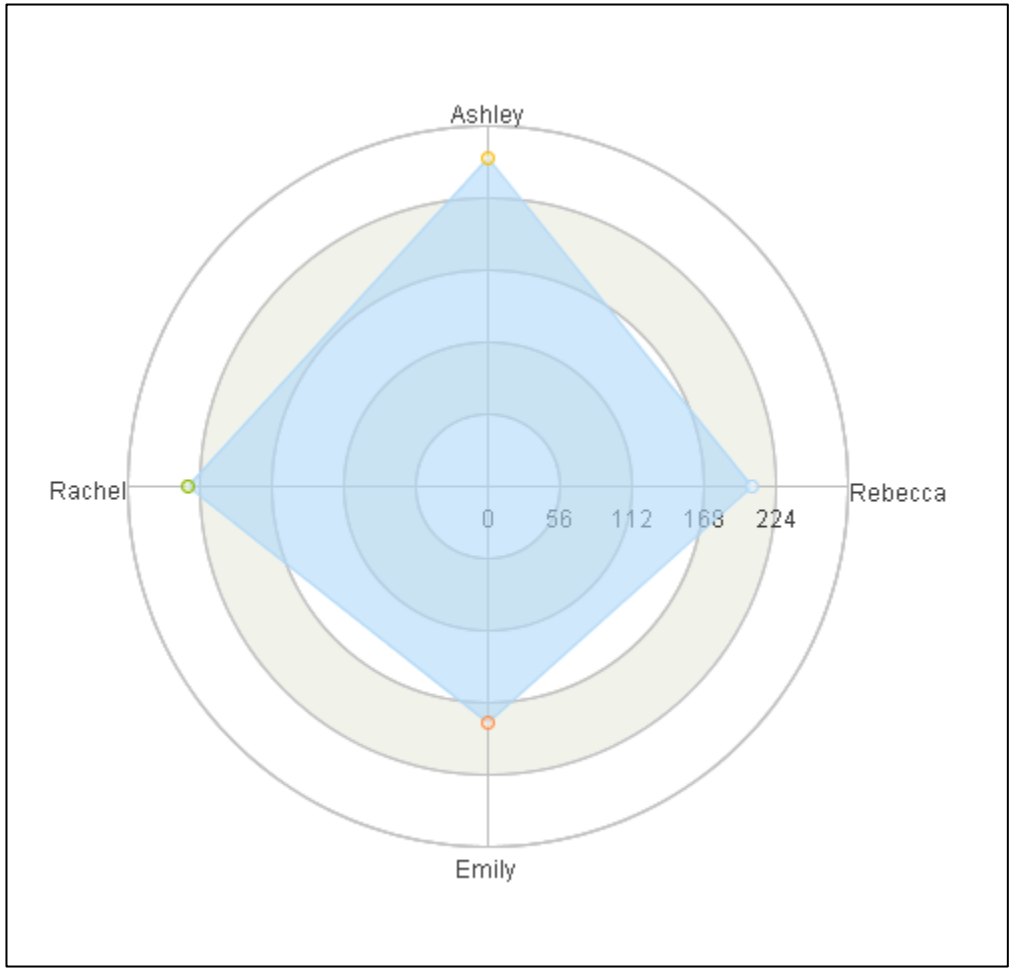
15.4.1 极坐标系

设置属性完成后的脚本如下：

	A	B
1	=file("summary.txt").import@t()	/导入总分表
2	=canvas()	/创建画布
3	=A2.plot("EnumAxis","name":"a","location":4)	/枚举型角轴，命名为a轴
4	=A2.plot("NumericAxis","name":"p","location":3)	/数值型极轴，命名为p轴
5	=A2.plot("Line","endToHead":true,"closedArea":true,"transparent":0.6,"axis1":"a","data1":A1.(NAME),"axis2":"p","data2":A1.(TOTAL))	/在画布中布置Line图元，设置数据且关联坐标系
6	=A2.draw(600,480)	/将画布按照指定大小绘制图形

15.4.1 极坐标系

执行后查看A6中的雷达图：



15.4.1 极坐标系

【例15.12】构造极坐标系，并将总分表(summary.txt)的数据使用**扇图元(Sector)**绘制成**饼图**。

绘制过程同样简单，只要构造极坐标系，然后布置扇图元即可。

扇图元的数据属性如下：

Edit Sector properties

Property	Value	Exp.
Text color		
Data		
Axis1	p	
Data1		A1.(NAME)
Axis2	a	
Data2		A1.(TOTAL)

注意：饼图的极轴只能为枚举轴。

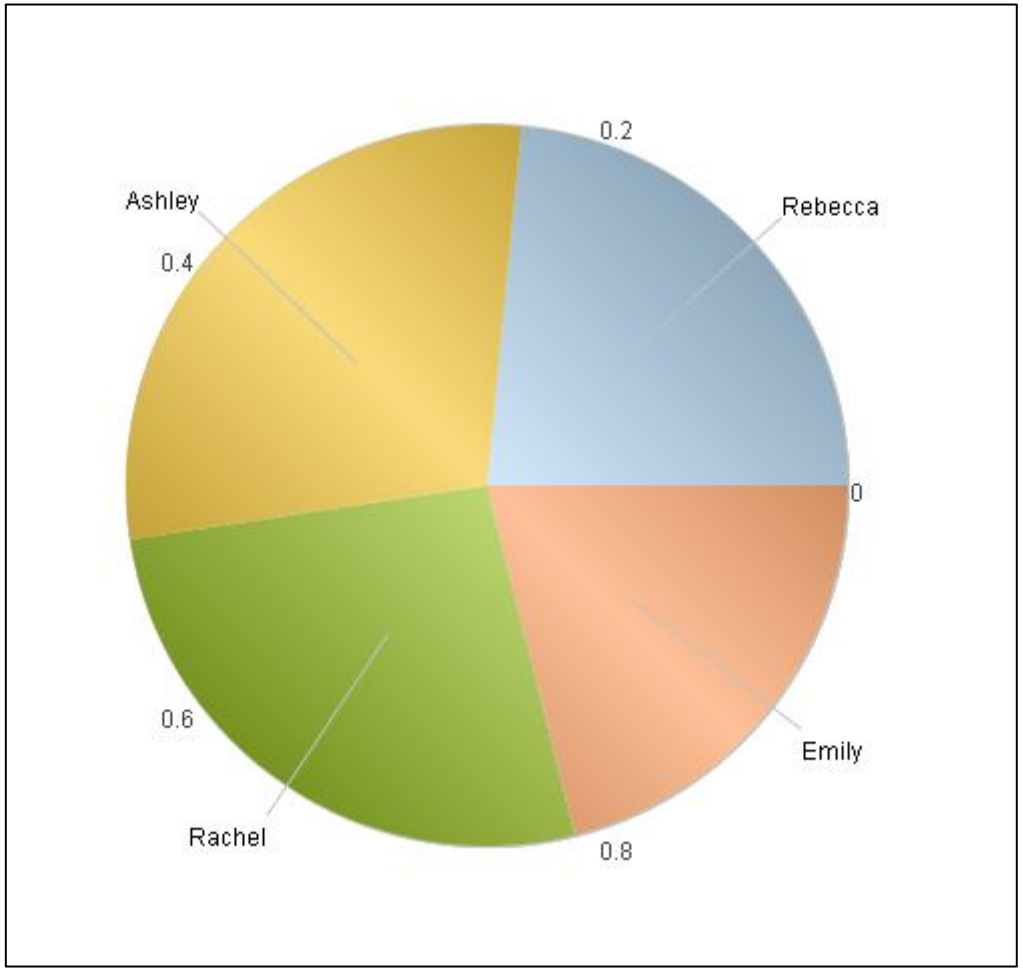
15.4.1 极坐标系

编辑好的示例脚本如下：

	A	B
1	=file("summary.txt").import@t()	/导入总分表
2	=canvas()	/创建画布
3	=A2.plot("EnumAxis","name":"p","location":3)	/枚举型极轴，命名为p轴
4	=A2.plot("NumericAxis","name":"a","location":4)	/数值型角轴，命名为a轴
5	=A2.plot("Sector","axis1":"p","data1":A1.(NAME),"axis2":"a","data2":A1.(TOTAL))	/在画布中布置Sector图元，设置数据且关联坐标系
6	=A2.draw(600,480)	/将画布按照指定大小绘制图形

15.4.1 极坐标系

执行后查看A6中的**饼图**：

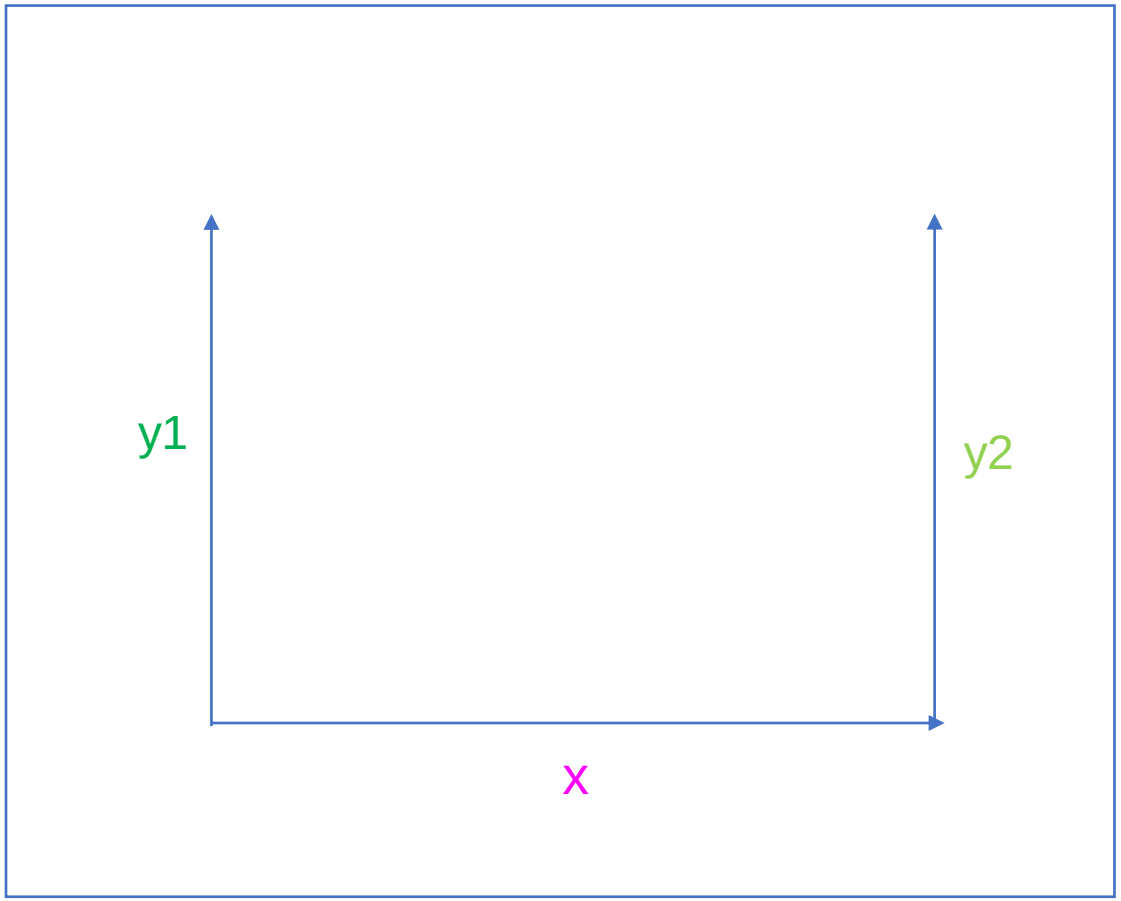


15.4.2 组合坐标系

前面所有示例都只用到两个轴组合的坐标系。为了绘制组合图元，可以构造包含公共轴的**组合坐标系**。

组合坐标系能方便地绘制出**双轴图**。

如右图布置好坐标轴后，便可以分别**在坐标系xy1**和**坐标系xy2**下绘制包含不同图元的双轴图。



15.4.2 组合坐标系

设置轴的坐标时，要根据轴的**定位方式(Location)**设置轴的相应物理坐标属性。

位置与坐标属性对应关系如图：

横轴（Horizontal））

纵轴（Vertical））

极轴（Polar））

角轴（Angle ））

15.4.2 组合坐标系

【例15.13】将分数表画成**柱形图**，然后将总分表用右边轴画成**折线图**，以生成**双轴柱线图。**

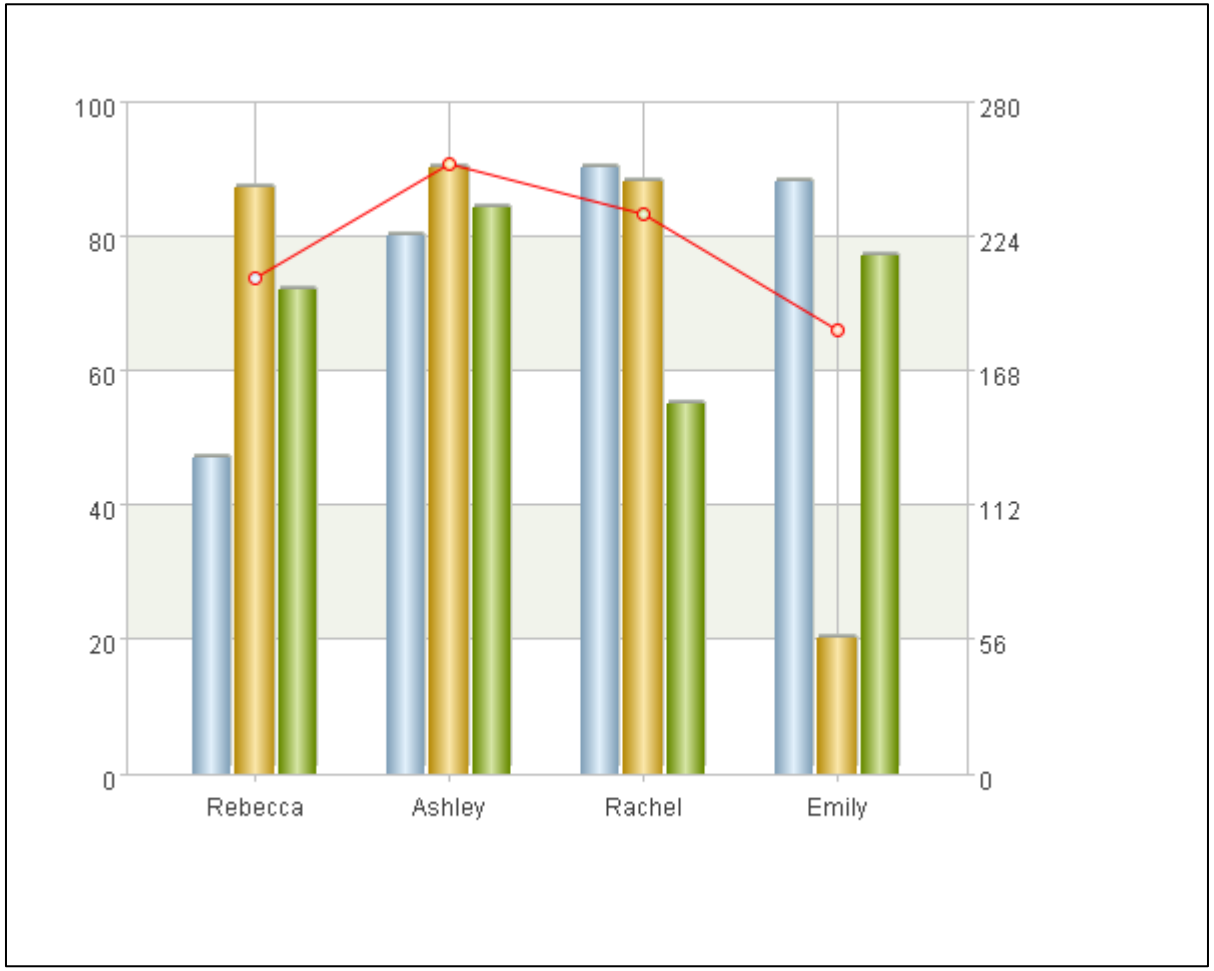
按照前面轴的示意图，设置好三个坐标轴，然后分别用不同图元使用各自的坐标系绘出图形便可。

属性的设置不再赘述，编辑好的脚本示例参考右表：

	A	B
1	=file("summary.txt").import@t()	/导入总分表
2	=file("scores.txt").import@t()	/导入分数表
3	=A2.pivot@r(ID,NAME;Subject,Score)	/分数表转置
4	=A3.(NAME+","+Subject)	/拼出分类和系列
5	=canvas()	/创建画布
6	=A5.plot("EnumAxis","name":"x")	/布置枚举横轴x
7	=A5.plot("NumericAxis","name":"y1","location":2)	/布置数值纵轴y1
8	=A5.plot("Column","axis1":"x","data1":A4,"axis2":"y1","data2":A3.(Score))	/在坐标系x-y1下画柱形图
9	=A5.plot("NumericAxis","name":"y2","location":2,"yPosition":0.8,"scalePosition":0)	/布置右边纵轴y2
10	=A5.plot("Line","lineColor":-65536,"axis1":"x","data1":A1.(NAME),"axis2":"y2","data2":A1.(TOTAL))	/坐标系x-y2下画折线图
11	=A5.draw(600,480)	/绘制双轴图

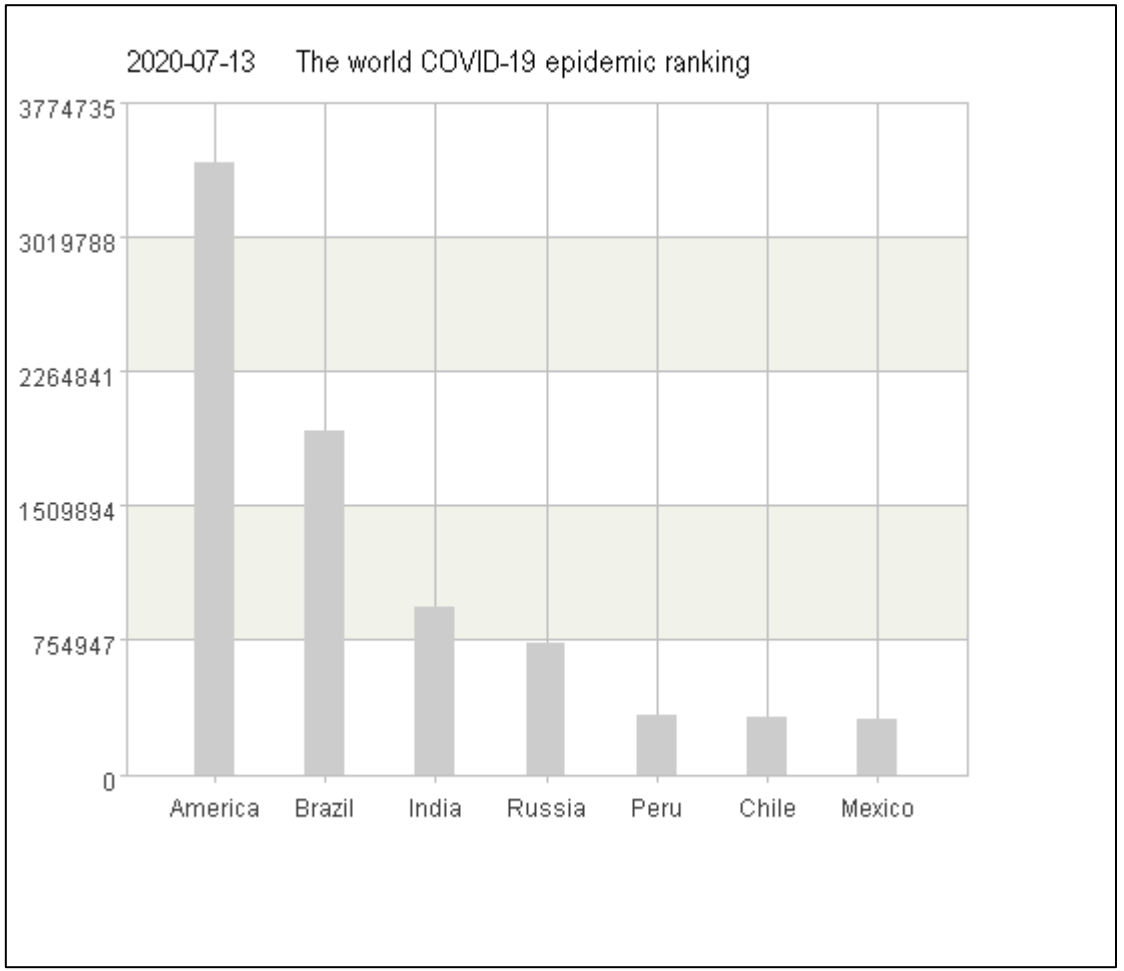
15.4.2 组合坐标系

执行后查看A11中的**双轴柱线图：**



15.4.3 图例（Legend）

查看右边的世界疫情图，每一根柱子由一对逻辑坐标（国家，累计病例数）确定。但如果想要再多表示一个逻辑量，比如根据病例数的多少来表示严重程度时，由于坐标系只有两个坐标量，此时就需要用颜色，或者形状来描述。



用颜色或形状等图形属性描述业务信息，需要建立业务信息和图形属性之间的对应关系，这就是图例(Legend)。

15.4.3 图例（Legend）

【例15.14】将2020年7月13日的世界疫情前7名，用柱状图画出来，并根据累计病例所在区间，用不同颜色标识出严重程度。

从网上摘抄到具体数据如下：

Country	Accumulation	Level
America	3431574	>1 Million
Brazil	1926824	>1 Million
India	936181	500 Thousand~1 Million
Russia	738787	500 Thousand~1 Million
Peru	333867	<500 Thousand
Chile	319493	<500 Thousand
Mexico	311486	<500 Thousand

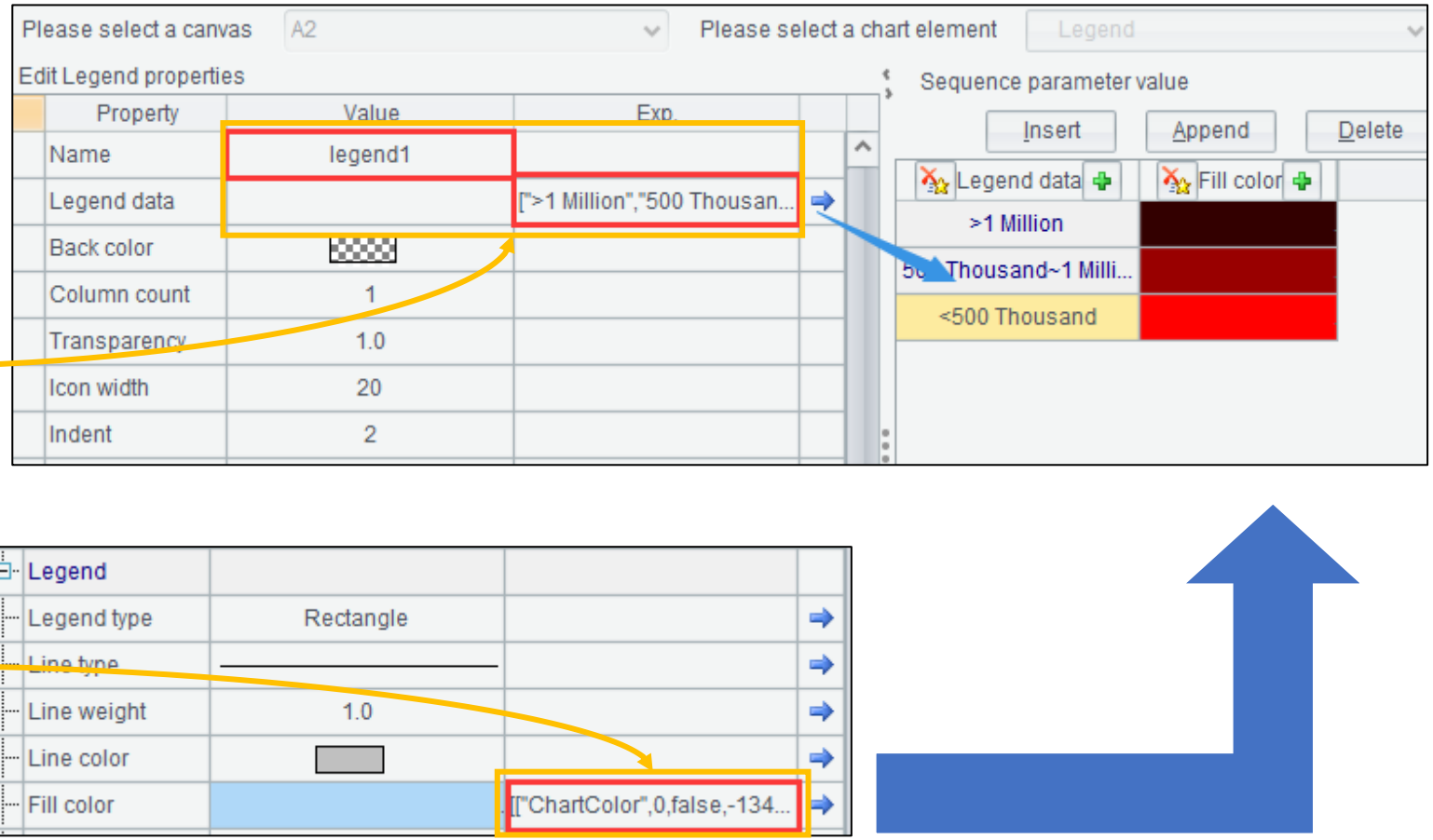
根据当前数据，需要用枚举轴表示Country。
用数值轴表示Accumulation。
然后再用图例绘制不同颜色来表示Level。

15.4.3 图例（Legend）

编辑图例，并如图设置图例的名字、图例数据(Legend data)和相应的颜色设置：

设置图例的名字为legend1以及图例数据

使用序列编辑模块设置累计病例人数区间对应的颜色)



15.4.3 图例（Legend）

编辑柱图元，直接设置柱图元的填充颜色为逻辑坐标Level，以及指定legend1，由legend1将逻辑值根据图例的设定转换为实际绘图颜色。

其他相关属性，以及属性对应的数据示意图：

Country	Accumulation	Level
America	3431574	>1 Million
Brazil	1926824	>1 Million
India	936181	500 Thousand~1 Milli..
Russia	738787	500 Thousand~1 Milli..
Peru	333867	<500 Thousand
Chile	319493	<500 Thousand
Mexico	311486	<500 Thousand

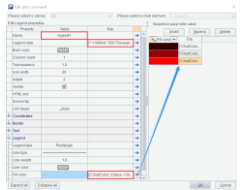
Edit Column properties

Property	Value	Exp.	Legend
Border weight	0		
Border color			
Fill color		A1.(Level)	legend1
Text			
Horizontal align...	Center		
Vertical alignm...	Top		
Text			
Font			
Text style	Example		
Text size	12		
Text color			
Data			
Axis1	x		
Data1		A1.(Country)	
Axis2	y		
Data2		A1.(Accumulation)	

经由图例legend1转换

逻辑颜色值

<500 Thousand



填充颜色值

["ChartColor",0,false,-65536,-5252872,0]

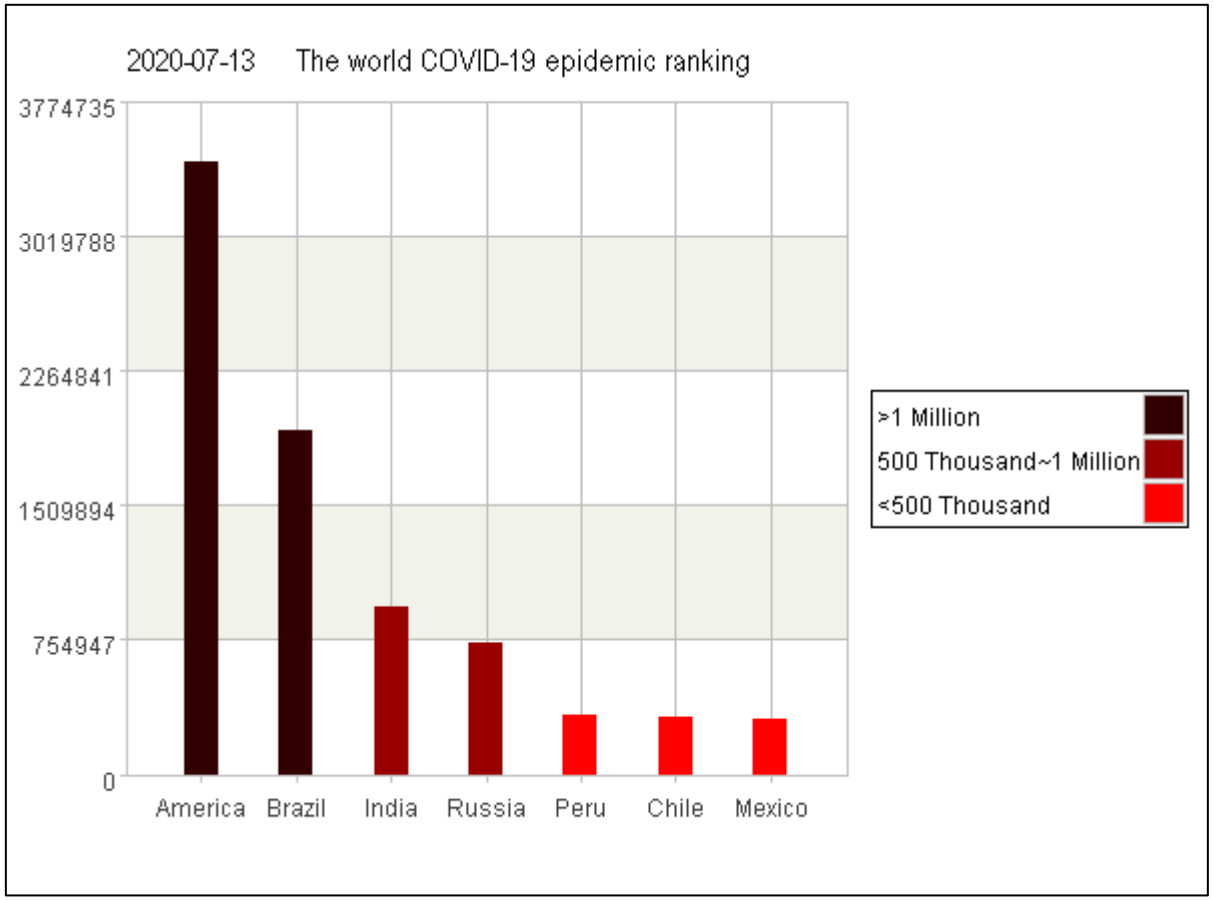
15.4.3 图例（Legend）

编辑好的示例脚本如下：

	A	B
1	=file("covid.txt").import@t()	/1.导入数据
2	=canvas()	/2.创建画布
3	=A2.plot("Legend","name":"legend1","legendText":[">1 Million","500 Thousand~1 Million","<500 Thousand"],"edgeIndent":2,"x":0.72,"y":0.4,"legendFillColor":[["ChartColor",0,false,-13434880,-5252872,0],["ChartColor",0,false,-6750208,-5252872,0],["ChartColor",0,false,-65536,-5252872,0]])	/3.设置图例的名字，值等相关属性
4	=A2.plot("EnumAxis","name":"x","xEnd":0.7)	/4.枚举型横轴x
5	=A2.plot("NumericAxis","name":"y","location":2)	/5.数值型纵轴y
6	=A2.plot("Column","shadow":false,"fillColor":A1.(Level):"legend1","axis1":"x","data1":A1.(Country),"axis2":"y","data2":A1.(Accumulation))	/6.布置柱图元
7	=A2.plot("Text","text":"2020-07-13 The world COVID-19 epidemic ranking","x":0.1,"y":0.03,"textFont":"Arial Narrow","textSize":16,"hAlign":0,"vAlign":8)	/7. 设置图形标题
8	=A2.plot("BackGround")	/缺省白色背景
9	=A2.draw@p(600,480)	/绘制图形

15.4.3 图例（Legend）

执行后查看A9中的**世界疫情柱图：**



15.4.3 图例（Legend）

使用颜色区别的图例在电脑上看，或者彩色打印出来时，没有问题。但如果打印成黑白色，彩色都变成了灰度颜色，不太好区分。此时可以使用点图，通过点的形状来设置图例。

【例15.15】将学生分数表(scores.txt)的科目按照不同点类型画成黑白点形图。

Edit Legend properties

Property	Value	Exp.
Name	legend	
Legend data		["Math","Physics","Chemistry"]

编辑图例，设置**图例类型(Legend type)**为**点(Dot)**，并设置相关的透明填充，黑色边线。

点形状(Marker shape)为跟图例名对应的形状序列。

Text		
Font		
Text style	Example	
Text size	12	
Text color		
Legend		
Legend type	Dot	
Line type		
Line weight	1.0	
Line color		
Fill color		
Marker shape	○	[1,2,3]

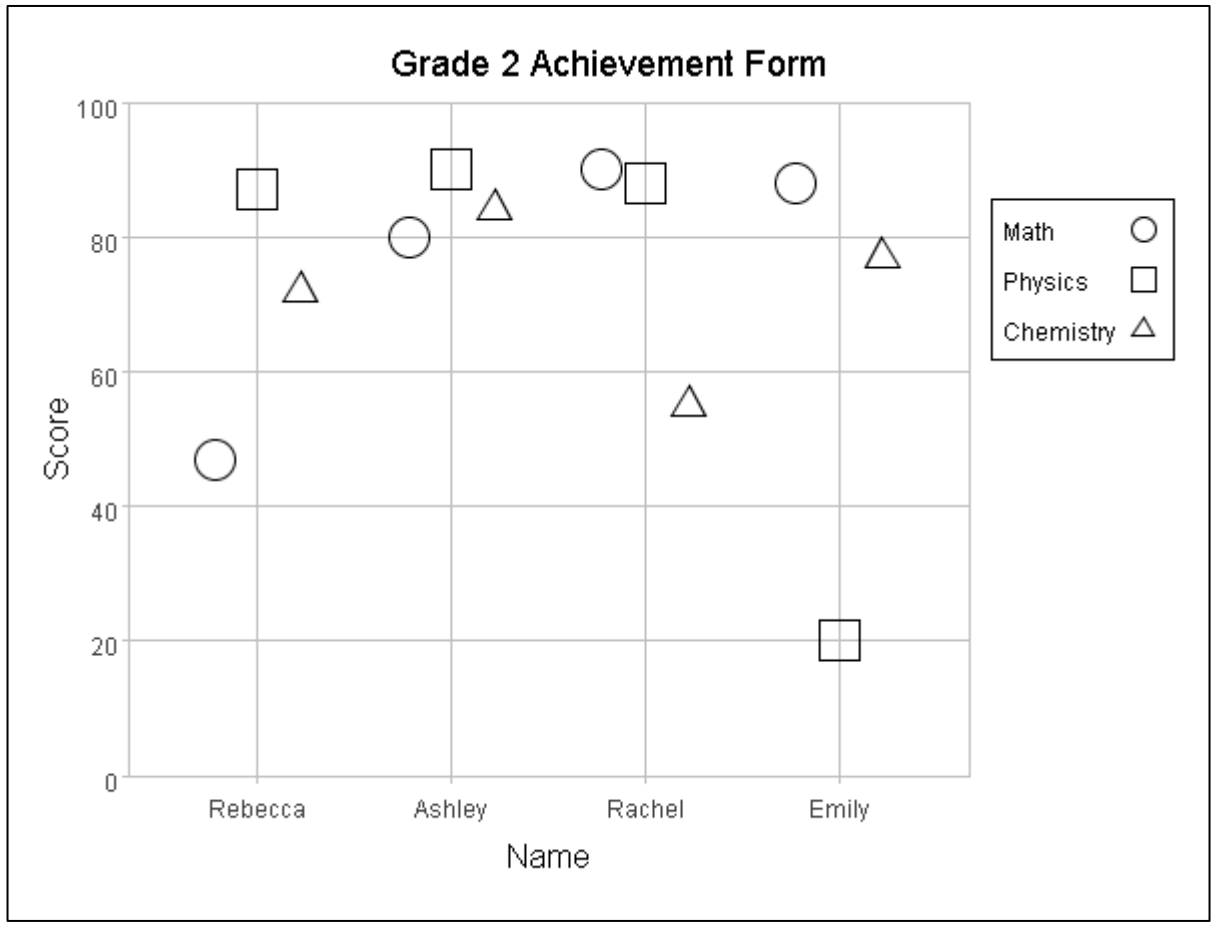
15.4.3 图例（Legend）

编辑好的示例脚本如下：

	A	B
1	=file("scores.txt").import@t()	/导入分数表
2	=A1.pivot@r(ID,NAME;Subject,Score)	/将分数交叉表转置
3	=A2.(NAME+","+Subject)	/拼出分类和系列
4	=canvas()	/创建画布
5	=A4.plot("EnumAxis","name":"x","title":"Name","regionColors":["ChartColor",0,false,16777215,-5252872,0])	/枚举型横轴，命名为x轴，设置轴标题
6	=A4.plot("NumericAxis","name":"y","location":2,"title":"Score","titleAngle":-90,"regionColors":["ChartColor",0,false,16777215,-5252872,0])	/数值型纵轴，命名为y轴，设置轴标题以及旋转角度
7	=A4.plot("Legend","name":"legend","legendText":["Math","Physics","Chemistry"],"legendType":2,"legendLineColor":-16777216,"legendFillColor":["ChartColor",0,false,16777215,-5252872,0],"legendMarkerShape":[1,2,3])	/在画布中布置图例，设置图例数据和填充颜色
8	=A4.plot("Dot","markerStyle":A2.(Subject):"legend","lineColor":-16777216,"markerColor":["ChartColor",0,false,16777215,-5252872,0],"markerWeight":10,"axis1":"x","data1":A3,"axis2":"y","data2":A2.(Score))	/布置Dot图元，设置逻辑点形状，并关联图例
9	=A4.plot("Text","text":"Grade 2 Achievement Form","y":0.05,"textSize":18)	/用文本绘制统计图标题
10	=A4.plot("BackGround")	/设置缺省白色背景
11	=A4.draw@p(600,480)	/绘制图形

15.4.3 图例（Legend）

执行后得到A11中的**黑白点形图。**



15.4.3 图例（Legend）

如果图例直接使用当前坐标系中枚举轴的分类，或者系列来绘制。而且图元和图例都采用缺省的调色板颜色序列时，此时图例自动简化为最简单情形，图例数据能从枚举轴上自动获取，这样的图例仅仅需要编辑一个图例对象即可。

【例15.16】将学生总分表(summary.txt)画成带图例的柱状图。

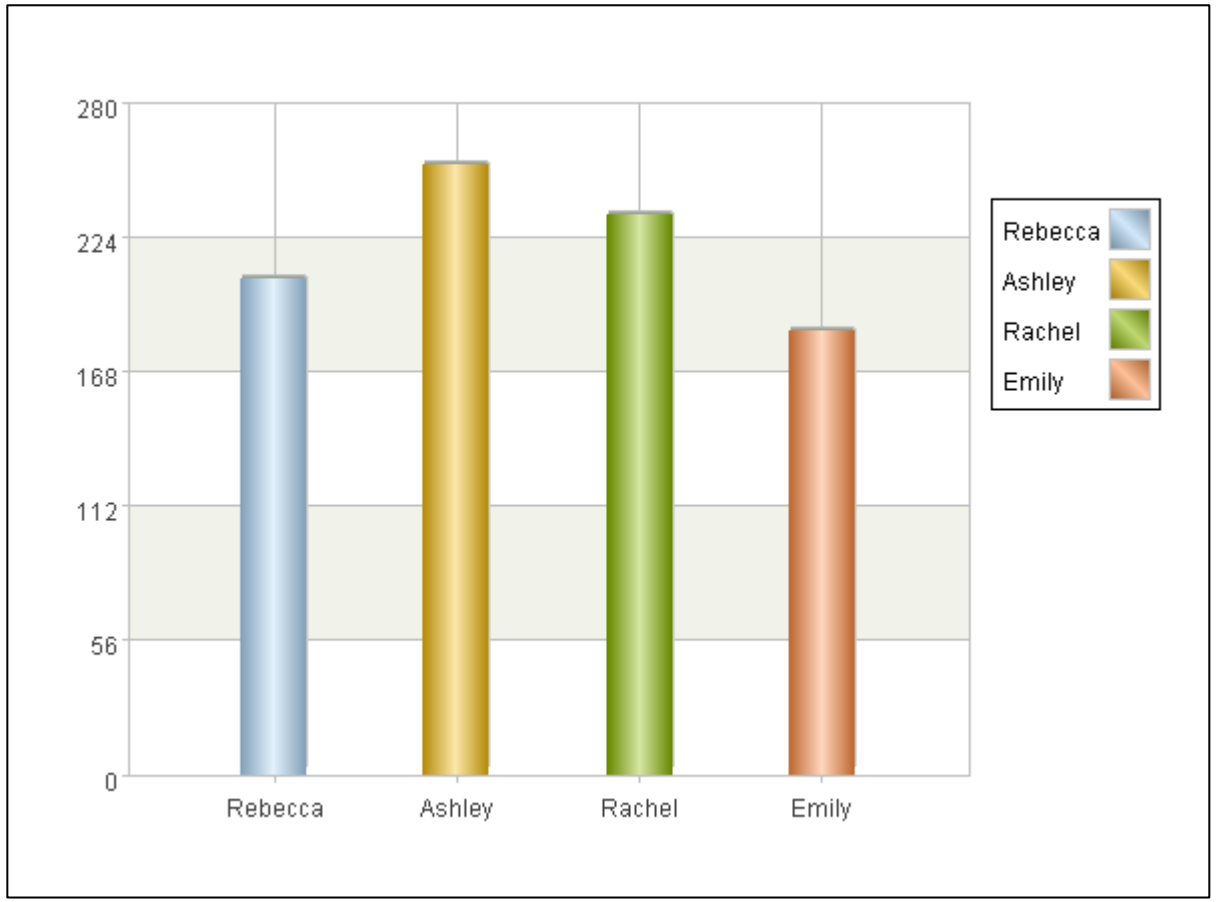
由于都是缺省属性，编辑非常简单，编辑好的示例脚本如右：

	A	B
1	=file("summary.txt").import@t()	/1.导入数据
2	=canvas()	/2.创建画布
3	=A2.plot("EnumAxis","name":"x")	/3.枚举型横轴x
4	=A2.plot("NumericAxis","name":"y","location":2)	/4.数值型纵轴y
5	=A2.plot("Column","axis1":"x","data1":A1.(NAME),"axis2":"y","data2":A1.(TOTAL))	/5.在画布中布置Column图元，设置数据且关联坐标系，使用缺省调色板
6	=A2.plot("Legend")	/6.布置Legend图例，同样使用缺省调色板
7	=A2.draw(600,480)	/7. 绘制图形

仅仅需要编辑一个图例对象即可，由于是自动匹配，名字都可以省略

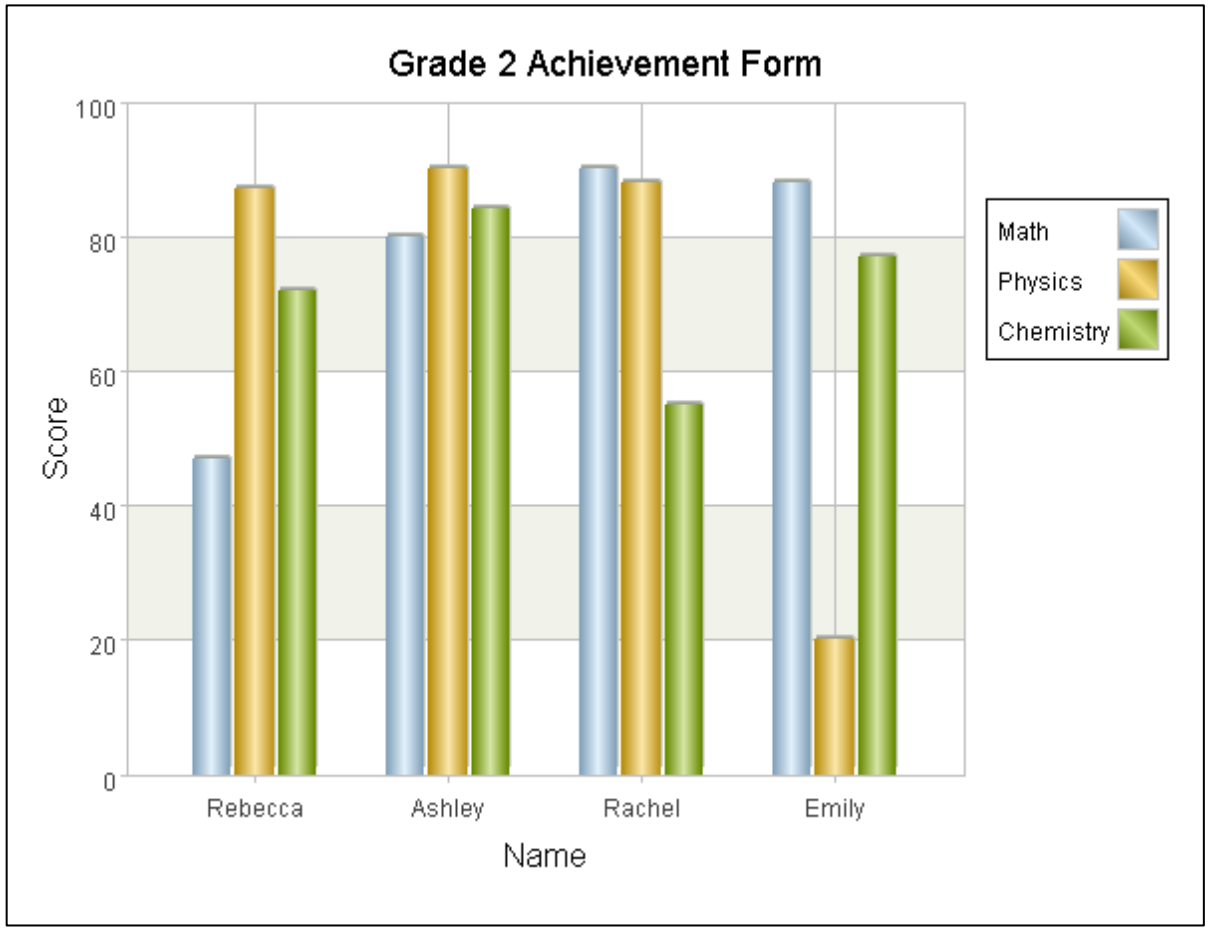
15.4.3 图例（Legend）

执行后查看A7中的**带图例柱图：**



15.4.3 图例（Legend）

回看【例15.9】，当枚举轴包含系列时，此时的图例会自动按照系列绘制：



15.5 成品图

15.5 成品图

使用图元绘制统计图，总是需要设定各种图形元素，以及坐标系，绘图过程比较繁琐。但有时候只需绘制单一图形，并不需要绘制灵活、复杂的统计图。

将几种常见统计图所需的各种图形元素，集成为一种**图形（Graph）**，这种封装好的单一图形称为**成品图**，后续都以图形来表示成品图。

图形的绘制只需指定**分类数据(Categories)**和**图形数据(Values)**，这两种数据的要求跟图元中的数据一致，也是两个序列。

15.5.1 柱状图(Column Graph)

柱状图（Column Graph）对应于柱图元，能将业务数据绘制为柱状图形。

【例15.17】将学生总分表(summary.txt)画成柱状图。

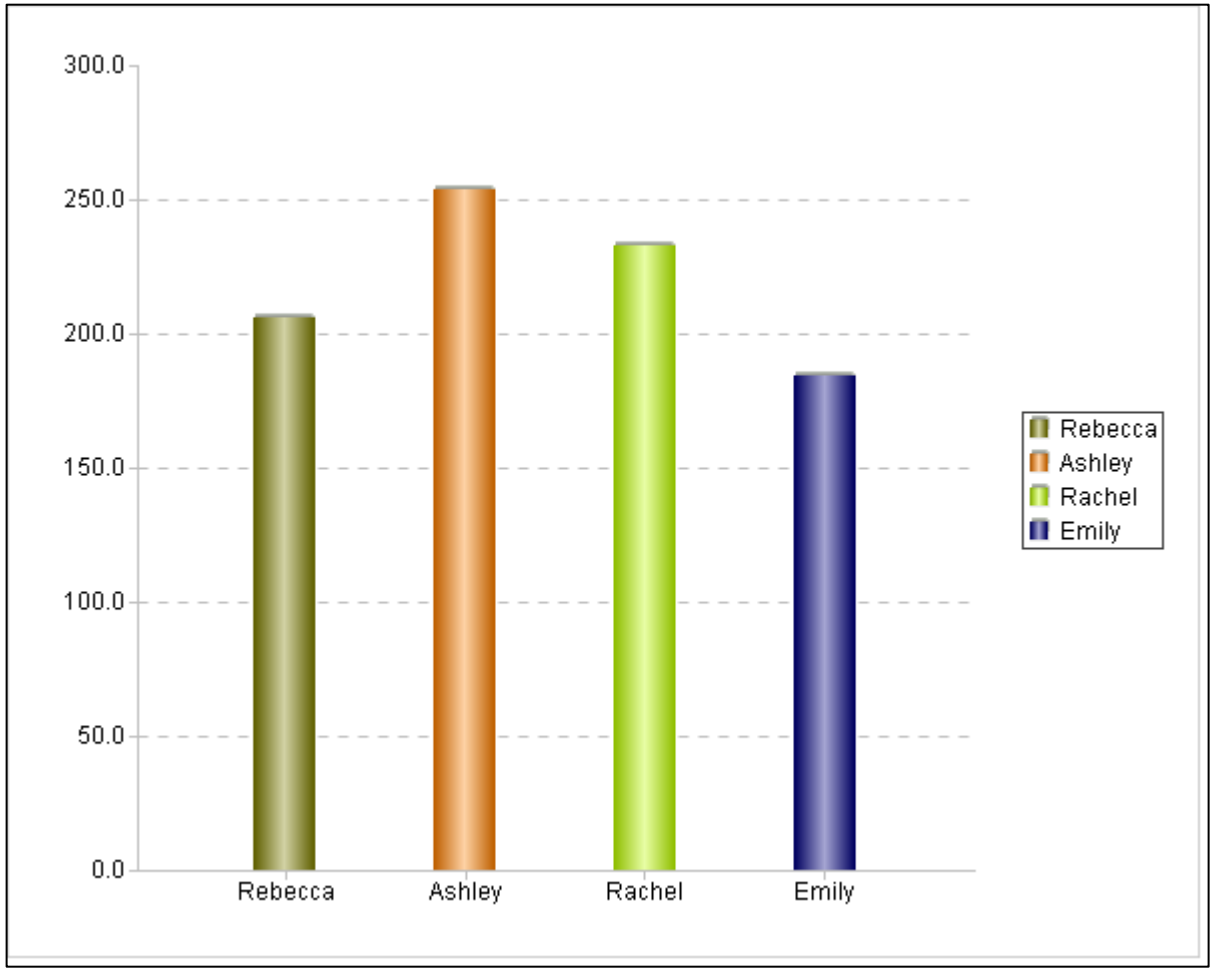
◆ 相较于图元的绘制，图形绘制更加简单，直接布置相应图形，指定业务数据便可绘图。

设置好属性后的示例脚本如下：

	A	B
1	=file("summary.txt").import@t()	/1.导入数据
2	=canvas()	/2.创建画布
3	=A2.plot("GraphColumn","categories":A1.(NAME),"values":A1.(TOTAL))	/3.往画布中添加Column Graph成品图形并设置数据
4	=A2.draw(600,480)	/4. 画布执行绘制

15.5.1 柱状图(Column Graph)

执行后查看A4中的**柱状图**：



15.5.2 双轴图(Dual-axis Graph)

双轴图应用也很广泛，同一图形中系列间的数值范围相差较大时，一般采用双轴图更直观。

【例15.18】计算出分数表（scores.txt）中每人的总分Total，然后使用**双轴图**，画出成绩为柱形，总分为折线的图形。

双轴图只能绘制包含多个系列的业务数据，由于有两个轴，因此需要指定具体的系列到相应轴。

如右图将计算列**总分（Total）**指定到右轴系列，则剩余系列都自动归为左轴系列：

Edit Dual-axis graph properties

Property	Value	Exp.
Categories		A4
Values		A3.(Score)
Type	2AxisColLine	
Left axis series		
Right axis series	Total	
Line properties		
Plot data points	☐	

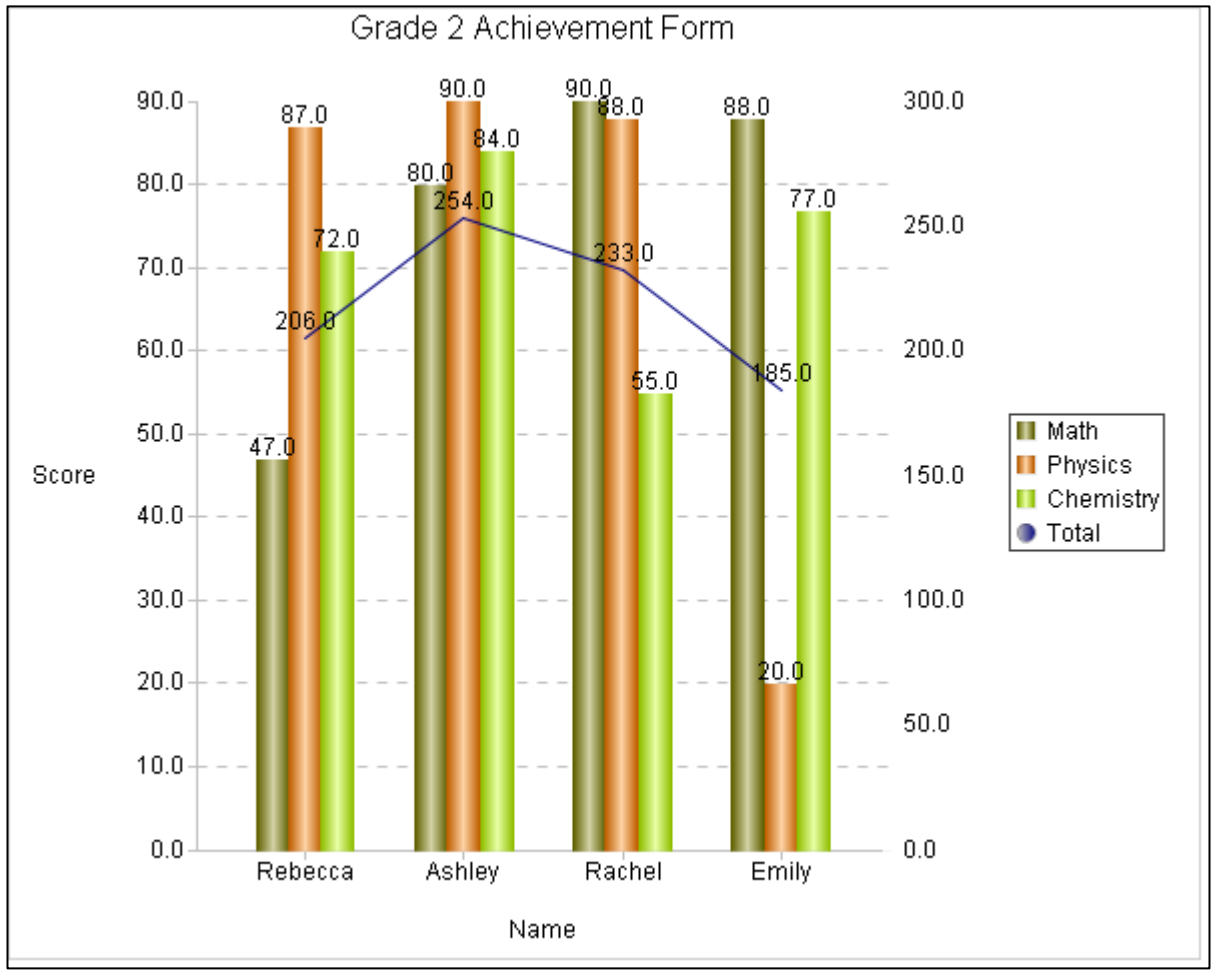
15.5.2 双轴图(Dual-axis Graph)

绘图示例脚本如下：

	A	B
1	=file("scores.txt").import@t()	/导入分数表
2	=A1.derive(Math+Physics+Chemistry:Total)	/增加计算列总分Total
3	=A2.pivot@r(ID,NAME;Subject,Score)	/将分数交叉表转置
4	=A3.(NAME+","+Subject)	/拼出分类和系列
5	=canvas()	/创建画布
6	=A5.plot("Graph2Axis","categories":A4,"values":A3.(Score),"rightSeries":"Total","xTitle":"Name","yTitle":"Score","graphTitle":"Grade 2 Achievement Form","displayData":2)	/在画布中布置Dual-axis graph图形并设置数据以及标题
7	=A5.draw(600,480)	/将画布按照指定大小绘制图形

15.5.2 双轴图(Dual-axis Graph)

执行后查看A7中的**双轴图**：



15.6 习题

15.6.1 习题：三角函数

【习题15.1】在直角坐标系下绘出余弦函数图。

解题思路：要绘制余弦函数图，只需要定义横轴和纵轴都为数值轴的坐标系。求出一段弧度内的余弦值后，再用折线图展现即可。

由于余弦值介于-1到1之间，因此纵轴要取消默认选中的属性**自动范围从0开始(AutoRange starts from 0)** 的选项。

纵轴设置属性如下图：

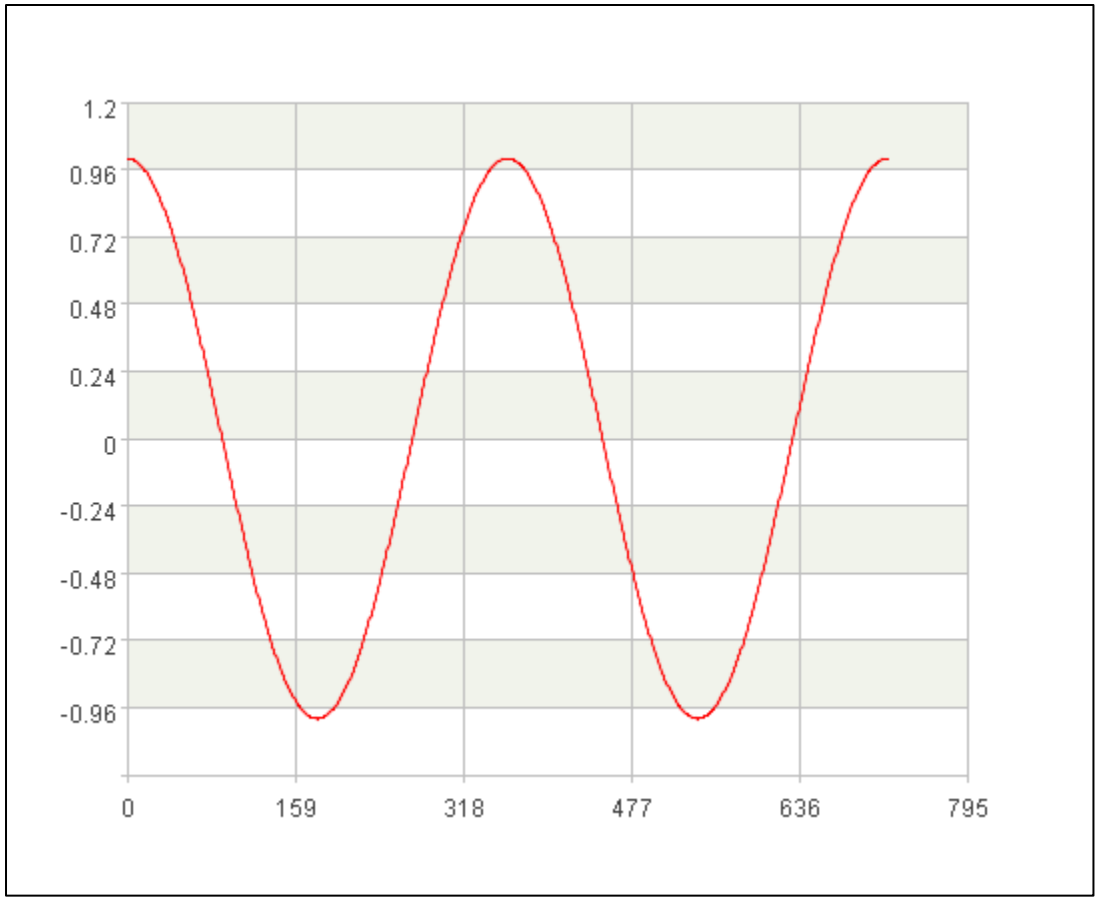
15.6.1 习题：三角函数

编辑好的示例脚本如下：

	A	B	C
1	>rads=[],angs=[]	/定义存储数据的序列	
2	for 720	=cos(A2*pi()/180)	/循环计算角度的余弦值
3		>angs=angs|A2	/构造角度序列
4		>rads=rads|B2	/构造余弦值序列
5	=canvas()	/创建画布	
6	=A5.plot("NumericAxis","name":"x")	/布置数值横轴x	
7	=A5.plot("NumericAxis","name":"y","location":2,"autoRangeFromZero":false)	/布置数值纵轴y，取消自动范围从0开始	
8	=A5.plot("Line","markerStyle":0,"lineColor":-65536,"axis1":"x","data1":angs,"axis2":"y","data2":rads)	/布置Line图元，设置坐标系和数据值	
9	=A5.plot("BackGround")	/布置缺省白色背景	
10	=A5.draw@p(600,480)	/绘制图形	

15.6.1 习题：三角函数

执行后查看A10中的**折线图**：



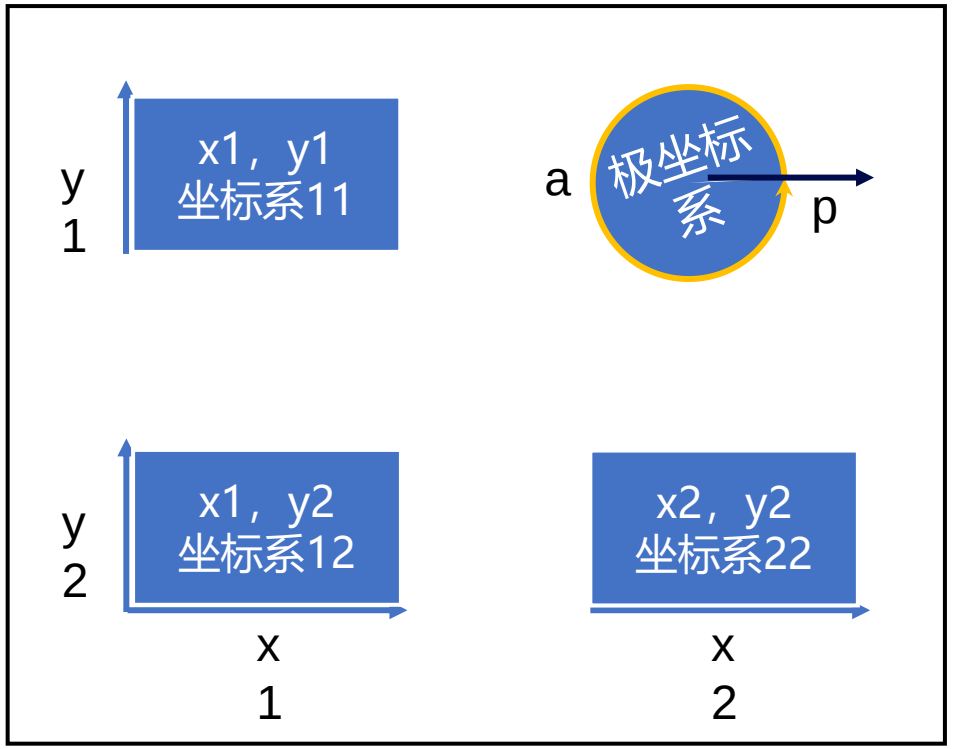
15.6.2 习题：多重组合

【习题15.2】将分数表(scores.txt)分别以**柱形图**、**堆积柱形图**置于同一图中，方便查看单科成绩以及总分排名。然后将总分表(summary.txt)以**扇形饼图**以及**折线图**也画在图中。

解题思路：灵活运用组合的多个坐标系，然后绘制相关图元，便可以画出复杂的多图。

由于是同一分数表分别绘制两种图元，所以可以采用组合图中的公共坐标系。

坐标系布局示意图如右图：



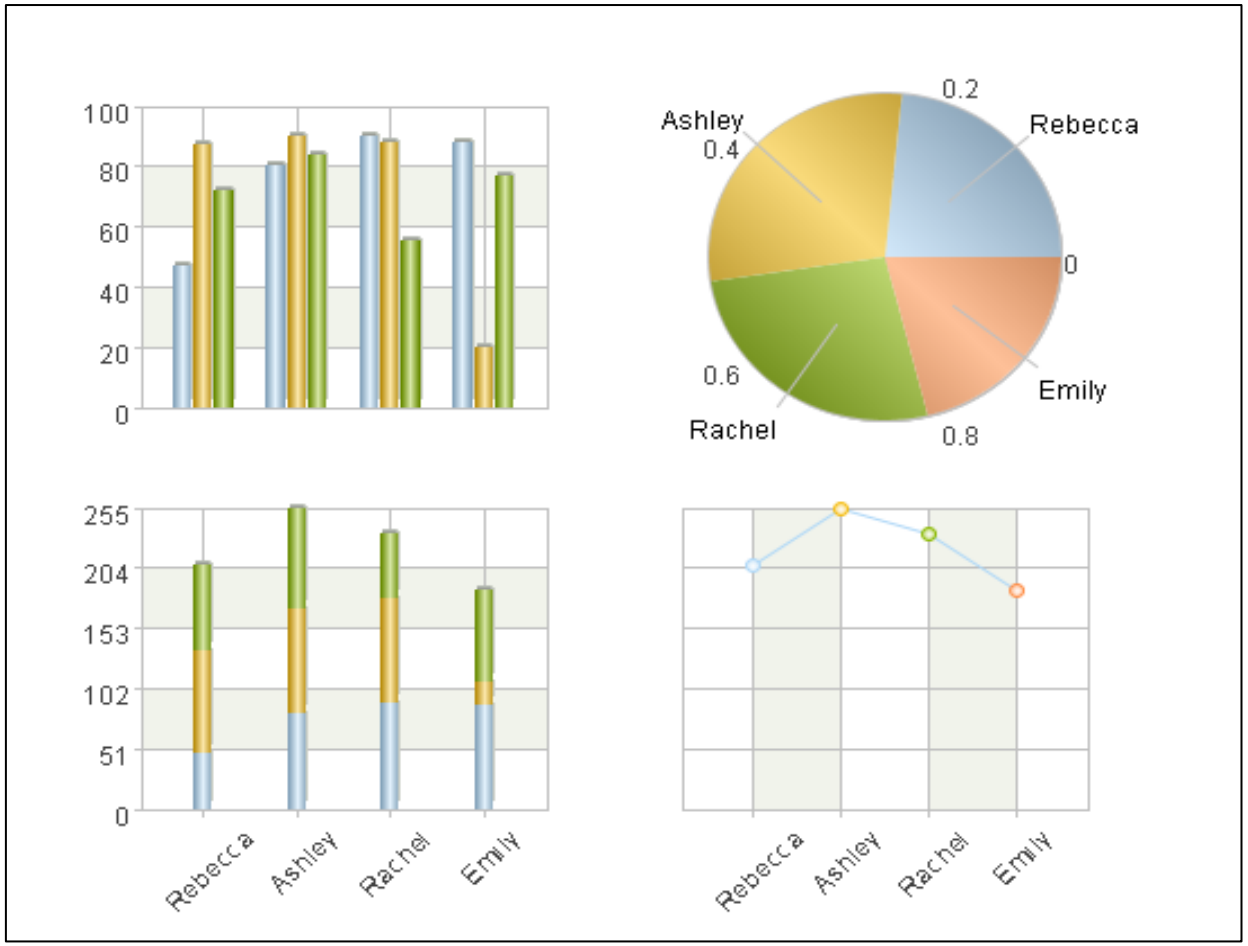
15.6.2 习题：多重组合

编辑好相应图元属性后的示例脚本如下：

	A	B	C	D
1	=file("summary.txt").import@t()	/导入总分表		
2	=file("scores.txt").import@t()	/导入分数表		
3	=A2.pivot@r(ID,NAME;Subject,Score)	/分数表转置		
4	=A3.(NAME+","+Subject)	/拼出分类和系列		
5	=canvas()	/创建画布		
6	=A5.plot("EnumAxis","name":"x1","xEnd":0.4,"labelAngle":-45,"labelOverlapping":true)	/布置枚举横轴x1	=A5.plot("EnumAxis","name":"p","location":3,"xEnd":0.4,"polarX":0.65,"polarY":0.25,"polarLength":0.13,"labelAngle":-45,"labelOverlapping":true)	/布置极轴p
7	=A5.plot("NumericAxis","name":"y1","location":2,"yStart":0.4,"yPosition":0.1)	/布置数值纵轴y1	=A5.plot("NumericAxis","name":"a","location":4)	/布置角轴a
8	=A5.plot("Column","axis1":"x1","data1":A4,"axis2":"y1","data2":A3.(Score))	/在坐标系11下画柱形图	=A5.plot("Sector","axis1":"p","data1":A1.(NAME),"axis2":"a","data2":A1.(TOTAL))	/极坐标系下画饼图
9	=A5.plot("NumericAxis","name":"y2","location":2,"yStart":0.8,"yEnd":0.5)	/布置纵轴y2，公用x1轴构建坐标系12	=A5.plot("EnumAxis","name":"x2","xStart":0.5,"labelAngle":-45,"labelOverlapping":true)	/布置横轴x2
10	=A5.plot("Column","stackType":2,"axis1":"x1","data1":A4,"axis2":"y2","data2":A3.(Score))	/坐标系12下画堆积柱图	=A5.plot("Line","axis1":"x2","data1":A1.(NAME),"axis2":"y2","data2":A1.(TOTAL))	/坐标系22下画折线
11	=A5.draw(600,480)	/绘制画布		

15.6.2 习题：多重组合

执行后查看A11中的**组合多图**：



15.6.3 习题：K线图

【习题15.3】绘制一个月内的股市K线图。

网上摘抄的部分数据如下：

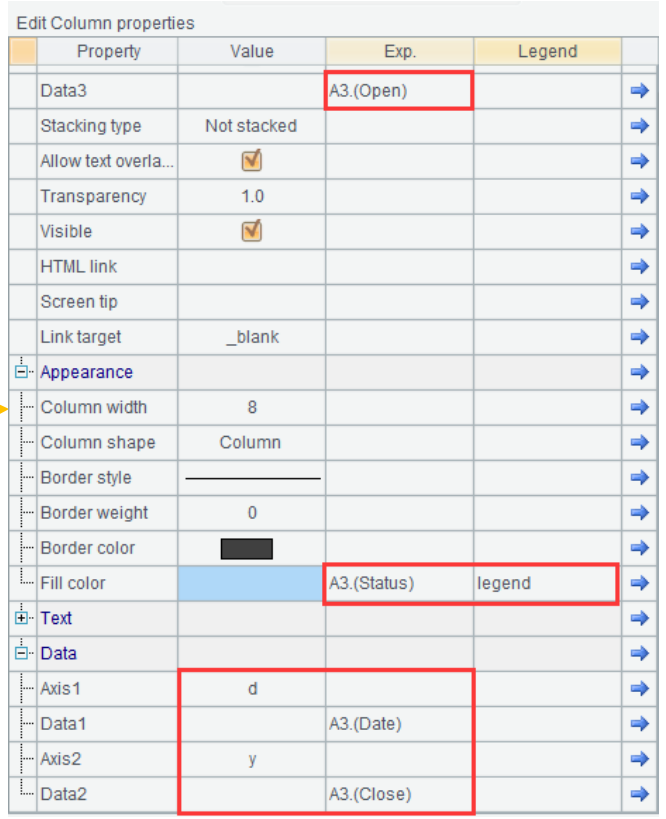
Date	Close	Vo...	Open	High	Low
07/13/2020	10390.84	0	10729.92	10824.78	10368.04
07/10/2020	10617.44	0	10545.91	10622.35	10447.01
07/09/2020	10547.75	0	10563.72	10578.1	10379.91
07/08/2020	10492.5	0	10409.35	10494.63	10350.96
07/07/2020	10343.89	0	10412.46	10518.98	10337.98
07/06/2020	10433.65	0	10360.38	10462.05	10354.98
07/02/2020	10207.63	0	10268.67	10310.36	10194.06
07/01/2020	10154.63	0	10063.67	10197.19	10048.04
06/30/2020	10058.77	0	9875.29	10085.59	9863.67
06/29/2020	9874.15	0	9771.72	9877.34	9663.61
06/26/2020	9757.22	0	9995.12	10000.67	9749.07
06/25/2020	10017.0	0	9899.36	10023.28	9810.47
06/24/2020	9909.17	0	10092.92	10137.5	9842.22
06/23/2020	10131.37	0	10130.83	10221.85	10112.44
06/22/2020	10056.47	0	9945.49	10059.61	9916.6
06/19/2020	9946.12	0	10042.13	10053.91	9872.94
06/18/2020	9943.05	0	9892.48	9959.2	9885.66

解题思路：K线图的横轴为日期，所以要用到**日期轴(Date axis)**作为横轴，然后每天有4个数据，开盘(Open)跟收盘(Close)为一组，用矩形表示；最低(Low)跟最高(High)为一组，用直线(也即宽度很窄的矩形)表示。**柱图元(Column)有Data3**属性，可以用于表示柱子的起始位置，从而满足要求。

构建直角坐标系，分别布置两个宽度不同，且使用**data3**的柱图元。柱子的颜色使用自定义图例来设置。

15.6.3 习题：K线图

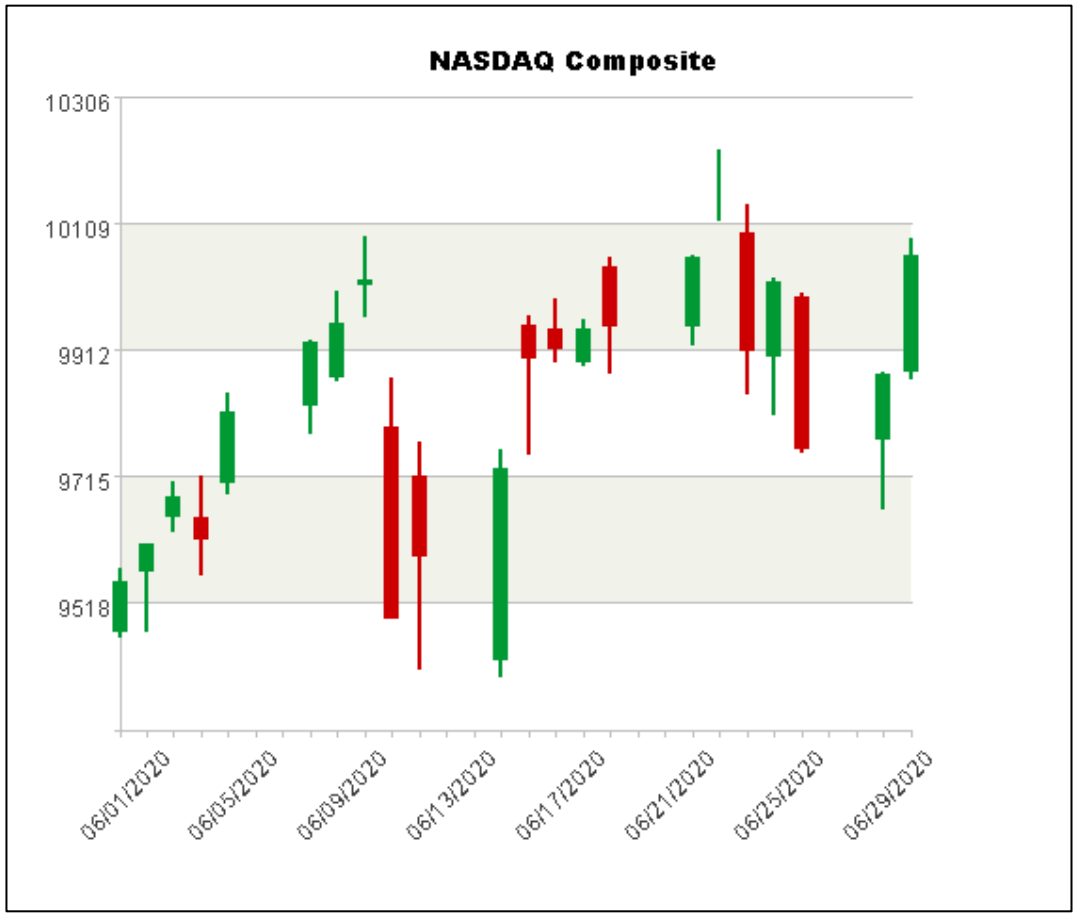
	A	B
1	=file("NASDAQ.csv").import@tc()	/导入数据
2	=A1.select(month(Date)==6)	/只画6月的数据
3	=A2.derive(Open>Close:Status)	/计算当天的涨跌
4	=canvas()	/创建画布
5	=A4.plot("Legend","name":"legend","legendText":[true,false],"visible":false,"legendFillColor":[["ChartColor",0,false,-3407872,-5252872,0],["ChartColor",0,false,-16737997,-5252872,0]])	/定义隐藏图例，用于柱图颜色映射
6	=A4.plot("DateAxis","name":"d","format":"MM/dd/yyyy","labelStep":3,"labelAngle":-45,"allowRegions":false)	/日期型横轴d
7	=A4.plot("NumericAxis","name":"y","location":2,"autoRangeFromZero":false)	/数值型纵轴y
8	=A4.plot("Column","shadow":false,"data3":A3.(Open),"columnWidth":8,"fillColor":A3.(Status):"legend","axis1":"d","data1":A3.(Date),"axis2":"y","data2":A3.(Close))	/柱图元，宽度为8像素的矩形，开盘收盘
9	=A4.plot("Column","shadow":false,"data3":A3.(Low),"columnWidth":2,"fillColor":A3.(Status):"legend","axis1":"d","data1":A3.(Date),"axis2":"y","data2":A3.(High))	/柱图元，宽度为2像素的矩形，最高最低
10	=A4.plot("Text","text":"NASDAQ Composite","y":0.05,"textFont":"Arial Black")	=A4.plot("BackGround")
11	=A5.draw@p(600,480)	/绘制图形



上图设置属性的含义为，在日期轴Data1处，数值轴从Data2到Data3处绘制柱子，填充色使用图例映射。

15.6.3 习题：K线图

执行后查看A11中的**柱图NASDAQ综合指数：**



第16章 大数据与游标

16.1 大数据与游标

- 大数据
- 游标概念

16.2 基础游标操作

- 用游标读取数据
- 循环游标

16.3 游标计算函数

- 游标的聚合计算
- 使用游标函数
- 延迟计算与立即计算

16.4 有序游标

16.5 大游标

- 大排序
- 大分组
- 合并
- 归并
- 连接
- 多路游标

16.6 习题

16.1 大数据与游标

16.1.1 大数据

我们之前讲的所有运算都在是内存中进行的，但有时候数据量会大到**不能全部加载进内存**，之前所讲的办法就不能直接用了

我们把这样的数据称为**大数据**，一般都以文件或数据库形式存放在硬盘上，本章讨论以文件形式存储的大数据上的计算

16.1.1 大数据

大数据就发生在我们的日常生活中：

某电商订单表

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	78002	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
3	123314	F	193846.25	1993/10/14	5-LOW	Clerk#000000955	0
4	136777	O	32151.78	1995/10/11	5-LOW	Clerk#000000124	0
5	44485	F	144659.2	1994/7/30	5-LOW	Clerk#000000925	0
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED	Clerk#000000058	0
7	39136	O	252004.18	1996/1/10	2-HIGH	Clerk#000000470	0
...	...	...	...	...	...	...	...

订单记录包含了订单号、客户号、订单状态、订单总价、订单日期、订单优先级等相关信息

随着用户与业务的扩大，数据量每年**成倍增长**

16.1.1 大数据

通话记录表

SERIALNUMBER	OUTID	INID	CALLTIME	DURATION	CHARGE
1	13781722635	18635273625	2006-08-11 08:53:22	97	0.00
2	18572635487	18173625183	2006-08-11 09:01:03	108	0.15
3	18572635182	13526384957	2006-08-11 09:02:11	34	0.00
4	13782769374	18502938475	2006-08-11 09:24:21	87	0.00
5	13672394817	18591820385	2006-08-11 09:33:12	132	0.15
6	18638274625	18108273645	2006-08-11 09:35:37	876	0.00
7	18127364918	13582937465	2006-08-11 09:43:18	56	0.15
…	…	…	…	…	…

通话记录包含了序号、拨打者、接听者、拨打时间、通话时长、费用等信息

手机用户可能上亿，可想而知实际通话记录信息的规模之大

16.1.2 游标概念

处理**大数据**计算时，无法将数据全部读入内存，只能分批读取，我们将这种读入数据的机制称为**游标**

游标可以方便地让我们使用循环处理小数据的方法来完成大数据的计算

16.1.2 游标概念

游标类似于一个指针，通过移动指针的位置来获取数据

游标一般具有单向性，也就是只能向前读取或跳过数据，不可回头。

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	78002	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
3	123314	F	193846.25	1993/10/14	5-LOW	Clerk#000000955	0
4	136777	O	32151.78	1995/10/11	5-LOW	Clerk#000000124	0
5	44485	F	144659.2	1994/7/30	5-LOW	Clerk#000000925	0
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED	Clerk#000000058	0
7	39136	O	252004.18	1996/1/10	2-HIGH	Clerk#000000470	0
...	...	...	...	...	...	...	...

16.2 基础游标操作

16.2.1 使用游标读取数据

【例16.1】使用游标读取订单表文件的**前5条**订单记录，订单表文件为orders.txt：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW
2	78002	O	46929.18	1996/12/1	1-URGENT
3	123314	F	193846.25	1993/10/14	5-LOW
4	136777	O	32151.78	1995/10/11	5-LOW
5	44485	F	144659.2	1994/7/30	5-LOW
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED
7	39136	O	252004.18	1996/1/10	2-HIGH

注：例题目的在于理解原理及函数用法，使用的数据量较小。下文的例题也有类似情况，不再说明。

16.2.1 使用游标读取数据

【例16.1】使用游标读取订单表文件的**前5条**订单记录，订单表文件为orders.txt：

解题思路

用f.cursor对文件创建游标，若文件第一行是字段名，用@t选项。用cs.fetch读取数据。

	A	B
1	=file("orders.txt")	/打开文件名为orders.txt的文件
2	=A1.cursor@t()	/对文件创建游标，这里的用法和import类似，也可以处理csv后缀的文件
3	=A2.fetch(5)	/从游标中读出前5条记录
4	=A2.close()	/游标中的数据没有全部读取完，需要关闭游标

A3返回的结果：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW
2	78002	O	46929.18	1996/12/1	1-URGENT
3	123314	F	193846.25	1993/10/14	5-LOW
4	136777	O	32151.78	1995/10/11	5-LOW
5	44485	F	144659.2	1994/7/30	5-LOW

16.2.1 使用游标读取数据

【例16.2】使用游标，**跳过前两条记录，**读取剩余订单表文件的订单记录，订单表文件为orders.csv：

```
O_ORDERKEY,O_CUSTKEY,O_ORDERSTATUS,O_TOTALPRICE,O_ORDERDATE,O_ORDERPRIORITY
1,36901,O,173665.47,1996/1/2,5-LOW
2,78002,O,46929.18,1996/12/1,1-URGENT
3,123314,F,193846.25,1993/10/14,5-LOW
4,136777,O,32151.78,1995/10/11,5-LOW
5,44485,F,144659.2,1994/7/30,5-LOW
6,55624,F,58749.59,1992/2/21,4-NOT SPECIFIED
7,39136,O,252004.18,1996/1/10,2-HIGH
```

16.2.1 使用游标读取数据

【例16.2】使用游标，跳过前两条记录，读取剩余订单表文件的订单记录，订单表文件为orders.csv：

解题思路

用cs.skip跳过记录。用cs.fetch读取数据。

	A	B
1	=file("orders.csv")	/打开文件名为orders.csv的文件
2	=A1.cursor@tc(;)	/对csv文件创建游标
3	=A2.skip(2)	/从当前游标跳过两条记录
4	=A2.fetch()	/从游标中读出剩余记录

A4返回的结果为：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3	123314	F	193846.25	1993/10/14	5-LOW
4	136777	O	32151.78	1995/10/11	5-LOW
5	44485	F	144659.2	1994/7/30	5-LOW
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED
7	39136	O	252004.18	1996/1/10	2-HIGH

注意：例题中共7条记录，所以使用了cs.fetch()直接读取剩余的所有记录。实际运用需要考虑数据量，避免内存溢出。

cs.skip(n;x) 详细说明：跳过n条记录或至x有变化，返回实际跳过的记录数。参数全部省略时返回剩余记录数并关闭游标。

16.2.1 使用游标读取数据

【例16.3】使用游标**过滤**订单表文件中客户号为**36901**的订单记录，订单表文件为orders.txt：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW
2	78002	O	46929.18	1996/12/1	1-URGENT
3	123314	F	193846.25	1993/10/14	5-LOW
4	136777	O	32151.78	1995/10/11	5-LOW
5	44485	F	144659.2	1994/7/30	5-LOW
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED
7	39136	O	252004.18	1996/1/10	2-HIGH

16.2.1 使用游标读取数据

【例16.3】使用游标过滤订单表文件中客户号为36901的订单记录

解题思路

用cs.select给游标附加过滤操作

	A	B
1	=file("orders.txt")	/打开文件名为orders.txt的文件
2	=A1.cursor@t()	/对文件创建游标
3	=A2.select(O_CUSTKEY==36901)	/在游标上添加过滤运算
4	=A2.fetch()	/获取游标运算的结果
5	=A2.fetch(1)	/获取游标的1条记录

A4返回的结果：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW

注意：因为游标具有单向性，执行完A4后，游标已经关闭，A5的返回结果为空。

16.2.2 循环游标

循环游标是指循环过程中，按照一定的规则从游标中读取一条或多条记录

例如：求某班级期末考试数学分数的**总成绩**

STU_ID	STU_NAME	MATH	ART	PHY	MATH_ACCUM
1	James	87	…	…	87
2	Jack	79	…	…	166
3	Alice	96	…	…	262
4	Andy	100	…	…	362
5	Lucy	54	…	…	316
6	Danny	69	…	…	385
7	Frank	80	…	…	465
…	…	…	…	…	…

游标**从头开始，向前移动**，每次读取（满足条件的）记录，进行计算。

16.2.2 循环游标

【例16.4】统计订单表中订单金额的**总和**，订单表文件为orders.txt：

O_ORDERKEY		O_CUSTKEY	O_ORDERSTATUS		O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW		
2	78002	O	46929.18	1996/12/1	1-URGENT		
3	123314	F	193846.25	1993/10/14	5-LOW		
4	136777	O	32151.78	1995/10/11	5-LOW		
5	44485	F	144659.2	1994/7/30	5-LOW		
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED		
7	39136	O	252004.18	1996/1/10	2-HIGH		

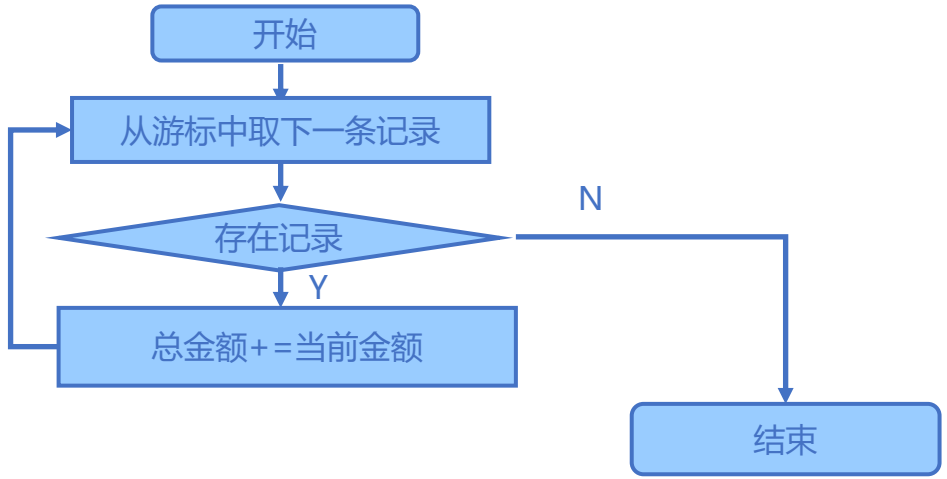
16.2.2 循环游标

【例16.4】统计订单表中订单金额的**总和**，订单表文件为orders.txt：

解题思路

用for cs语句循环读取游标中的数据，并按要求进行计算。

	A	B	C
1	=file("orders.txt").cursor@t(O_TOTALPRICE:decimal)	0	/A1打开文件并创建游标，这里仅用金额字段，并设定其类型为decimal，B1初值0
2	for A1	>B1+=A2.O_TOTALPRICE	/A2循环游标A1，每次取1条记录，B2将A2的金额累加至B1



B1
Value
902005.65

需要注意的是，在大数据计算中，**双精度数据的精度往往不够**，此时可以用**deciaml**类型计算。

16.2.2 循环游标

循环游标还可以每次读取指定条数据，进行计算

例如：对某订单表数据，每次读取10000条记录，统计客户号为4的订单总金额

游标

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	78002	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
...	...	...	...	...	...	...	...
10001	62017	O	225029.87	1998/6/28	4-NOT SPECIFIED	Clerk#000000177	0
10002	114073	F	87784.83	1995/1/30	2-HIGH	Clerk#000000498	0
...	...	...	...	...	...	...	...
20001	139027	F	167087.15	1992/5/6	4-NOT SPECIFIED	Clerk#000000121	0
20002	60173	O	142775.84	1995/12/15	5-LOW	Clerk#000000268	0
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...

10000条

10000条

10000条

客户号为4的订单总额：2648536.79

订单金额求和：118436.64

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
6477	4	O	31172.78	1992/1/2	5-LOW	Clerk#000000154	0
8794	4	O	87263.86	1995/12/1	1-URGENT	Clerk#000000271	0

订单金额求和：1131.2

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
18582	4	F	1131.2	1996/5/3	4-NOT SPECIFIED	Clerk#000000618	0

订单金额求和：0

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY

16.2.2 循环游标

【例16.5】统计订单表中客户号（O_CUSTKEY）为4的订单金额（O_TOTALPRICE）的总和，订单表文件为orders.txt：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	4	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
…	…	…	…	…	…	…	…
10001	62017	O	225029.87	1998/6/28	4-NOT SPECIFIED	Clerk#000000177	0
10002	114073	F	87784.83	1995/1/30	2-HIGH	Clerk#000000498	0
…	…	…	…	…	…	…	…
20001	4	F	167087.15	1992/5/6	4-NOT SPECIFIED	Clerk#000000121	0
20002	60173	O	142775.84	1995/12/15	5-LOW	Clerk#000000268	0
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…

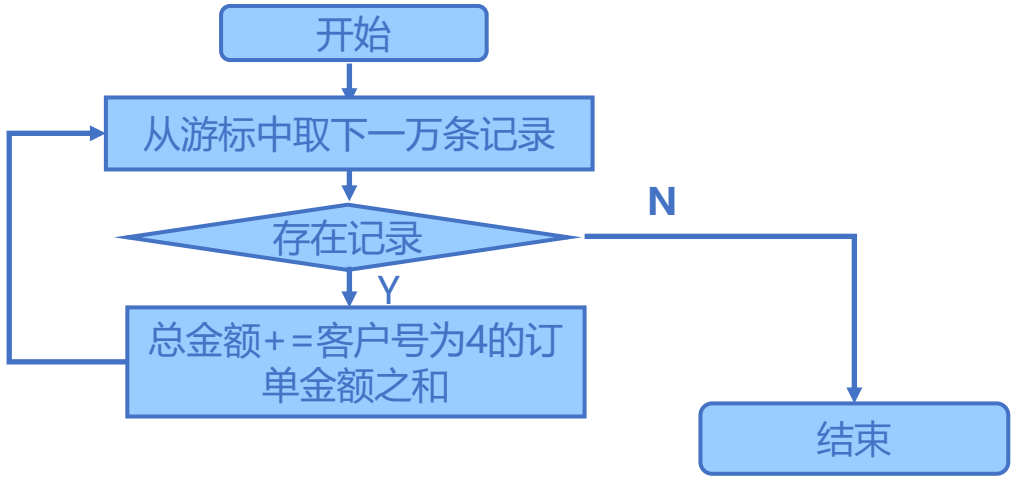
16.2.2 循环游标

【例16.5】统计订单表中客户号（O_CUSTKEY）为4的订单金额（O_TOTALPRICE）的总和，订单表文件为orders.txt：

解题思路

用for cs,n语句来循环读取游标中的数据，对每次读出的n条数据，按要求进行计算。

	A	B	C
1	=file("orders.txt").cursor@t(O_CUSTKEY,O_TOTALPRICE:decimal)	0	/A1打开文件并创建游标，B1初值0
2	for A1,10000	=A2.select(O_CUSTKEY==4).sum(O_TOTALPRICE)	/A2循环游标，每次10000条记录，B2对这些数据过滤与计算
3		>B1=B1+B2	/B3将B2计算出的结果累加到B1



B1
Value
2648536.79

16.2.2 循环游标

循环游标时，有时并不会遍历所有数据

例如：找到客户号（O_CUSTKEY）**为4的前三条记录。**

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	78002	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
3	4	F	311722.87	1992/4/26	3-MEDIUM	Clerk#000000361	0
4	18273	O	13827.11	1993/8/6	1-URGENT	Clerk#000000280	0
5	4	O	277493.04	1995/11/1	2-HIGH	Clerk#000000465	0
6	73652	F	9273.5	1994/6/2	5-LOW	Clerk#000000618	0
7	4	O	143971.54	1996/6/3	1-URGENT	Clerk#000000978	0
…	…	…	…	…	…	…	…

满足条件停止循环，第7条记录后的数据**不再读取**

16.2.2 循环游标

【例16.6】查找订单表中客户号（O_CUSTKEY）**为4的前三条记录**，订单表文件为orders.txt：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	78002	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
…	…	…	…	…	…	…	…
164711	4	F	311722.87	1992/4/26	3-MEDIUM	Clerk#000000361	0
…	…	…	…	…	…	…	…
385825	4	O	277493.04	1995/11/1	2-HIGH	Clerk#000000465	0
…	…	…	…	…	…	…	…
1192231	4	O	143971.54	1996/6/3	1-URGENT	Clerk#000000978	0
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…

16.2.2 循环游标

【例16.6】查找订单表中客户号（O_CUSTKEY）**为4的前三条记录**，订单表文件为orders.txt：

解题思路

对循环读出的数据按要求进行计算，满足一定条件时停止循环

	A	B	C	D
1	=file("orders.txt").cursor@t()	[]		/A1打开文件并创建游标，B1设初值空集
2	for A1,10000	=A2.select(O_CUSTKEY==4)	>B1=B1|B2	/A2循环游标，每次10000条记录，B2取客户号为4的记录，C2将结果追加到B1
3		if B1.len()>=3	>B1=B1(to(3))	/B3判断是否已经取够三条，若满足条件，C3因为B1可能大于3条件记录，取其前三条
4			break	/若B3满足条件，停止循环
5	>A1.close()			/因数据可能没有全部读完，主动关闭游标

B1

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
164711	4	F	311722.87	1992/4/26	3-MEDIUM	Clerk#000000361	0
385825	4	O	277493.04	1995/11/1	2-HIGH	Clerk#000000465	0
1192231	4	O	143971.54	1996/6/3	1-URGENT	Clerk#000000978	0

例题中实际仅循环了30次游标，**远小于**全部遍历需要的循环次数。

16.3 游标计算函数

16.3.1 游标的聚合运算

上一节讲了用**循环游标**来处理游标上的聚合运算，但显然太麻烦了。

和序表类似，游标上也有封装好的**直接聚合运算。**

16.3.1 游标的聚合运算

【例16.7】订单表（orders.txt）中，统计所有订单的订单数量、最大订单金额（max(O_TOTALPRICE)）、订单总金额（sum(O_TOTALPRICE)）

订单表（orders.txt）部分数据如下：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	78002	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
3	123314	F	193846.25	1993/10/14	5-LOW	Clerk#000000955	0
4	136777	O	32151.78	1995/10/11	5-LOW	Clerk#000000124	0
5	44485	F	144659.2	1994/7/30	5-LOW	Clerk#000000925	0
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED	Clerk#000000058	0
7	39136	O	252004.18	1996/1/10	2-HIGH	Clerk#000000470	0
32	130057	O	208660.75	1995/7/16	2-HIGH	Clerk#000000616	0
33	66958	F	163243.98	1993/10/27	3-MEDIUM	Clerk#000000409	0

16.3.1 游标的聚合运算

【例16.7】订单表（orders.txt）中，统计所有订单的订单数量、最大订单金额（max(O_TOTALPRICE)）、订单总金额（sum(O_TOTALPRICE)）

解题思路

使用cs.total函数，完成全集的各类聚合

	A	B
1	=file("orders.txt").cursor@t()	/打开文件并创建游标
2	=A1.total(count(~),max(O_TOTALPRICE),sum(O_TOTALPRICE))	/统计订单总数、最大交易金额及总金额

A2：

Index	Member
1	1500000
2	555285.16
3	2.2682930644744 66E11

三行结果分别为：总订单数、最大交易金额、总金额

和序表的全集聚合运算（A.sum/A.count等）不同。因为游标只能遍历一次，要一次同时计算多个值出来，所以封装在同一个函数（cs.total）中。

16.3.1 游标的聚合运算

【例16.8】从订单表（orders.txt）中，统计各年（year(O_ORDERDATE)）金额（O_TOTALPRICE）最低和最高的三笔订单。

订单表（orders.txt）部分数据如下：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	78002	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
3	123314	F	193846.25	1993/10/14	5-LOW	Clerk#000000955	0
4	136777	O	32151.78	1995/10/11	5-LOW	Clerk#000000124	0
5	44485	F	144659.2	1994/7/30	5-LOW	Clerk#000000925	0
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED	Clerk#000000058	0
7	39136	O	252004.18	1996/1/10	2-HIGH	Clerk#000000470	0
32	130057	O	208660.75	1995/7/16	2-HIGH	Clerk#000000616	0
33	66958	F	163243.98	1993/10/27	3-MEDIUM	Clerk#000000409	0

16.3.1 游标的聚合运算

【例16.8】从订单表（orders.txt）中，统计各年（year(O_ORDERDATE)）金额（O_TOTALPRICE）最低和最高的三笔订单

解题思路

使用cs.groups函数分组统计，聚合函数用top

	A	B
1	=file("orders.txt").cursor@t()	/打开文件并创建游标
2	=A1.groups(year(O_ORDERDATE):Year;top(3;O_TOTALPRICE):Min,top(-3;O_TOTALPRICE):Max)	/按年份统计订单金额最低、最高的三笔订单记录，其中最高的三笔，在top函数中用负号

A2：

Index	Year	Min	Max
1	1992	[[1600323,53551,F, ...],[82...	[[1750466,21433,F, ...],[358...
2	1993	[[5267200,107698,F, ...],[4...	[[4515876,100685,F, ...],[38...
3	1994	[[5363650,1351,F, ...],[404...	[[4722021,128120,F, ...],[3...
4	1995	[[5062848,148522,F, ...],[1...	[[3967937,109186,O, ...],[57...
5	1996	[[1788706,120961,O, ...],[...	[[2199712,66790,O, ...],[215...
6	1997	[[3790757,85468,O, ...],[2...	[[3043270,144617,O, ...],[45...
7	1998	[[2159139,3115,O, ...],[51...	[[1395745,100159,O, ...],[33...

Index	O_ORDER...	O_CUSTKEY	O_ORDER...	O_TOTAL...	O_ORDER...	O_O
1	1600323	53551	F	866.9	1992-04...	1-U
2	823814	64043	F	870.88	1992-01...	1-U
3	3309383	82949	F	908.18	1992-07...	1-U

Index	O_ORDER...	O_CUSTKEY	O_ORDER...	O_TOTAL...	O_ORDER...	O_O
1	1750466	21433	F	555285.16	1992-11...	4-N
2	3586919	24049	F	522644.48	1992-11...	1-U
3	2185667	51796	F	511359.88	1992-10...	1-U

游标的聚合运算也可以使用在游标函数返回的游标上

16.3.1 游标的聚合运算

【例16.9】订单表（orders.txt）中，统计所有订单的订单数量、最大订单金额（max(O_TOTALPRICE)）、订单总金额（sum(O_TOTALPRICE)）

订单表（orders.txt）部分数据如下：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	78002	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
3	123314	F	193846.25	1993/10/14	5-LOW	Clerk#000000955	0
4	136777	O	32151.78	1995/10/11	5-LOW	Clerk#000000124	0
5	44485	F	144659.2	1994/7/30	5-LOW	Clerk#000000925	0
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED	Clerk#000000058	0
7	39136	O	252004.18	1996/1/10	2-HIGH	Clerk#000000470	0
32	130057	O	208660.75	1995/7/16	2-HIGH	Clerk#000000616	0
33	66958	F	163243.98	1993/10/27	3-MEDIUM	Clerk#000000409	0

16.3.1 游标的聚合运算

【例16.9】订单表（orders.txt）中，统计所有订单的订单数量、最大订单金额（max(O_TOTALPRICE)）、订单总金额（sum(O_TOTALPRICE)）

解题思路

使用cs.groups(;...)函数，完成全集的各类聚合

	A	B
1	=file("orders.txt").cursor@t()	/打开文件并创建游标
2	=A1.groups(;count(~):count,max(O_TOTALPRICE):max,sum(O_TOTALPRICE):total)	/统计订单总数、最大交易金额及总金额

A2：

Index	count	max	total
1	1500000	555285.16	2.2682930644744 66E11

cs.groups (;...)中的聚合函数，支持sum/count/max/min/top/avg

16.3.2 使用游标函数

类似地，**序表**相关的**过滤、对位产生、关联**等操作也有对应的**游标函数**。

但是，和**序表**不同的是，**游标**上的这些函数不能立即就得到结果，而是把这些操作附加在游标上，然后再返回这个游标，等以后读取游标数据时才真正执行这些操作。

16.3.2 使用游标函数

用游标函数**生成新数据（cs.new）**：

例如：对雇员表，将NAME与SURNAME字段合并后生**成新字段FULLNAME，去掉**原字段NAME和SURNAME。

游标

EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT
1	Rebecca	Moore	F	California	1974/11/20	2005/3/11	R&D
2	Ashley	Wilson	F	New York	1980/7/19	2008/3/16	Finance
3	Rachel	Johnson	F	New Mexico	1970/12/17	2010/12/1	Sales
4	Emily	Smith	F	Texas	1985/3/7	2006/8/15	HR
5	Ashley	Smith	F	Texas	1975/5/13	2004/7/30	R&D

游标

EID	FULLNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT
1	Rebecca Moore	F	California	1974/11/20	2005/3/11	R&D
2	Ashley Wilson	F	New York	1980/7/19	2008/3/16	Finance
3	Rachel Johnson	F	New Mexico	1970/12/17	2010/12/1	Sales
4	Emily Smith	F	Texas	1985/3/7	2006/8/15	HR
5	Ashley Smith	F	Texas	1975/5/13	2004/7/30	R&D

【例16.10】将雇员表中的名字（NAME）和姓氏（SURNAME）**合并为全名（FullName）**，雇员表文件为emp.txt：

EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT
1	Rebecca	Moore	F	California	1974/11/20	2005/3/11	R&D
2	Ashley	Wilson	F	New York	1980/7/19	2008/3/16	Finance
3	Rachel	Johnson	F	New Mexico	1970/12/17	2010/12/1	Sales
4	Emily	Smith	F	Texas	1985/3/7	2006/8/15	HR
5	Ashley	Smith	F	Texas	1975/5/13	2004/7/30	R&D
6	Matthew	Johnson	M	California	1984/7/7	2005/7/7	Sales
7	Alexis	Smith	F	Illinois	1972/8/16	2002/8/16	Sales
8	Megan	Wilson	F	California	1979/4/19	1984/4/19	Marketing
9	Victoria	Davis	F	Texas	1983/12/7	2009/12/7	HR
10	Ryan	Johnson	M	Pennsylvania	1976/3/12	2006/3/12	R&D
…	…	…	…	…	…	…	…

16.3.2 使用游标函数

【例16.10】将雇员表中的名字（NAME）和姓氏（SURNAME）**合并为全名（FullName）**，雇员表文件为emp.txt：

解题思路

使用游标函数cs.new，在原游标上生成新数据结构。

	A	B
1	=file("emp.txt").cursor@t(EID,NAME,SURNAME,GENDER,STATE,BIRTHDAY)	/打开文件并创建游标
2	=A1.new(EID,NAME+" "+SURNAME:FullName,GENDER,STATE,BIRTHDAY)	/将名字与姓氏合并为全名，去掉名字与姓氏原字段
3	=A2.fetch()	/读取记录

A3

Index	EID	FullName	GENDER	STATE	BIRTHDAY
1	1	Rebecca Moore	F	California	1974-11-20
2	2	Ashley Wilson	F	New York	1980-07-19
3	3	Rachel Johnson	F	New Mexico	1970-12-17
4	4	Emily Smith	F	Texas	1985-03-07
5	5	Ashley Smith	F	Texas	1975-05-13

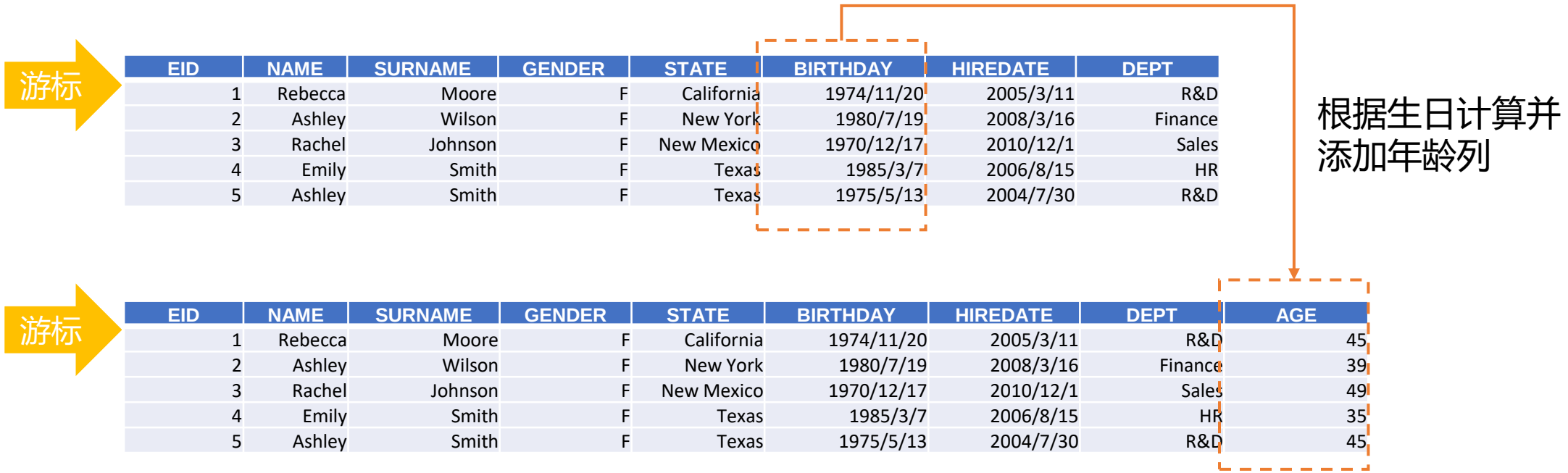
例题使用的数据量较小，fetch出了所有结果，实际运用中，**需要考虑是否会内存溢出**。

结果的前5条记录如图所示。

16.3.2 使用游标函数

用游标函数添加字段（cs.derive）：

例如：对雇员表，根据雇员生日（BIRTHDAY），**计算并添加**年龄（AGE）列。



16.3.2 使用游标函数

【例16.11】根据雇员表中雇员生日（BIRTHDAY），**计算并添加**年龄（AGE）列，雇员表文件为emp.txt：

EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT
1	Rebecca	Moore	F	California	1974/11/20	2005/3/11	R&D
2	Ashley	Wilson	F	New York	1980/7/19	2008/3/16	Finance
3	Rachel	Johnson	F	New Mexico	1970/12/17	2010/12/1	Sales
4	Emily	Smith	F	Texas	1985/3/7	2006/8/15	HR
5	Ashley	Smith	F	Texas	1975/5/13	2004/7/30	R&D
6	Matthew	Johnson	M	California	1984/7/7	2005/7/7	Sales
7	Alexis	Smith	F	Illinois	1972/8/16	2002/8/16	Sales
8	Megan	Wilson	F	California	1979/4/19	1984/4/19	Marketing
9	Victoria	Davis	F	Texas	1983/12/7	2009/12/7	HR
10	Ryan	Johnson	M	Pennsylvania	1976/3/12	2006/3/12	R&D
…	…	…	…	…	…	…	…

16.3.2 使用游标函数

【例16.11】根据雇员表中雇员生日（BIRTHDAY），计算并添加年龄（AGE）列，雇员表文件为emp.txt：

解题思路

使用游标函数cs.derive，在原游标上计算并添加列。

	A	B
1	=file("emp.txt").cursor@t(EID,NAME,SURNAME,GENDER,STATE,BIRTHDAY)	/创建文件游标，只读指定字段
2	=A1.derive(age(BIRTHDAY):Age)	/添加计算列Age，计算雇员年龄
3	=A2.fetch()	/读取原游标添加列后的数据

A3

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	Age
1	1	Rebecca	Moore	F	California	1974-11...	45
2	2	Ashley	Wilson	F	New York	1980-07...	39
3	3	Rachel	Johnson	F	New Mexico	1970-12...	49
4	4	Emily	Smith	F	Texas	1985-03...	35
5	5	Ashley	Smith	F	Texas	1975-05...	45

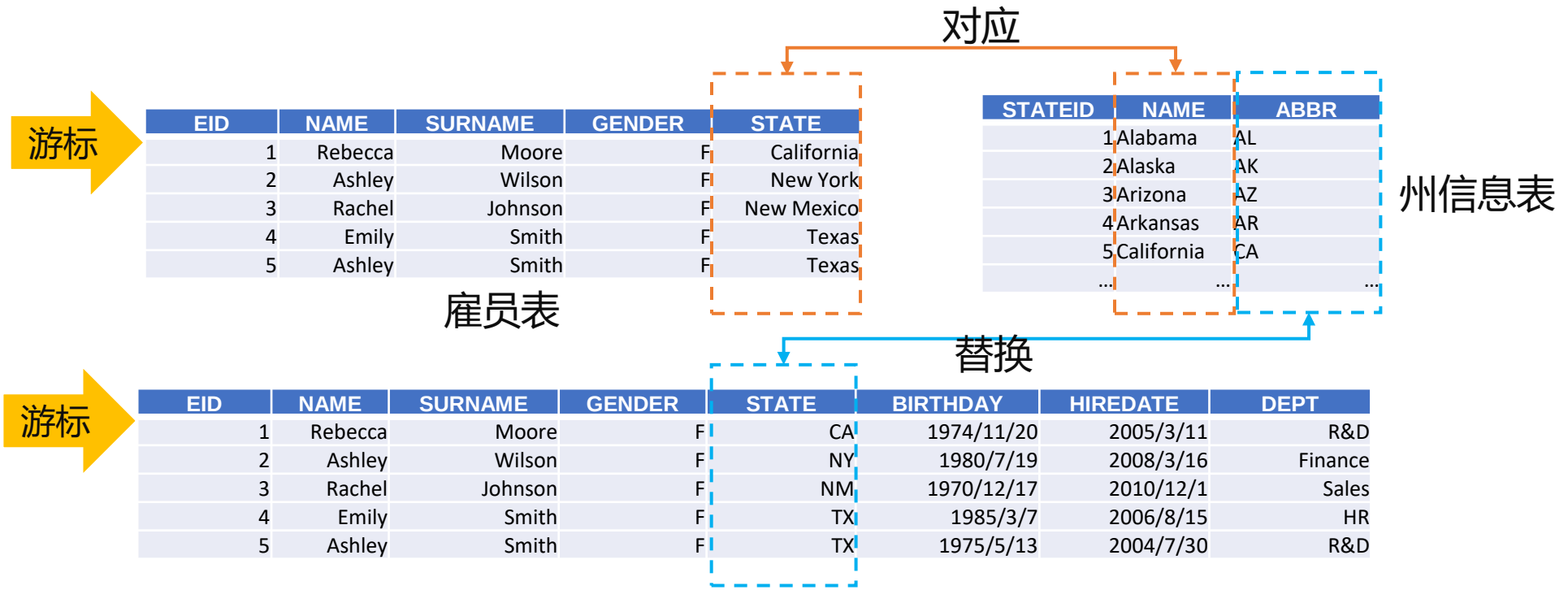
例题使用的数据量较小，fetch出了所有结果，实际运用中，需要考虑是否会内存溢出。

结果的前5条记录如图所示。

16.3.2 使用游标函数

用游标函数修改字段（cs.run）：

例如：将雇员表中所在州（SATAE），根据州信息表的州名（NAME）替换为简称（ABBR）



16.3.2 使用游标函数

【例16.12】将雇员表中所在州（SATAE），根据州信息表的州名（NAME）**替换为简称（ABBR）**，雇员表文件为emp.txt、州信息表states.txt：

EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT
1	Rebecca	Moore	F	California	1974/11/20	2005/3/11	R&D
2	Ashley	Wilson	F	New York	1980/7/19	2008/3/16	Finance
3	Rachel	Johnson	F	New Mexico	1970/12/17	2010/12/1	Sales
4	Emily	Smith	F	Texas	1985/3/7	2006/8/15	HR
5	Ashley	Smith	F	Texas	1975/5/13	2004/7/30	R&D
6	Matthew	Johnson	M	California	1984/7/7	2005/7/7	Sales
7	Alexis	Smith	F	Illinois	1972/8/16	2002/8/16	Sales
8	Megan	Wilson	F	California	1979/4/19	1984/4/19	Marketing
9	Victoria	Davis	F	Texas	1983/12/7	2009/12/7	HR
10	Ryan	Johnson	M	Pennsylvania	1976/3/12	2006/3/12	R&D
…	…	…	…	…	…	…	…

雇员表（emp.txt）

STATEID	NAME	ABBR
1	Alabama	AL
2	Alaska	AK
3	Arizona	AZ
4	Arkansas	AR
5	California	CA
6	Colorado	CO
7	Connecticut	CT
8	Delaware	DE
9	Florida	FL
10	Georgia	GA
…	…	…

州信息表（states.txt）

16.3.2 使用游标函数

【例16.12】将雇员表中所在州（SATAE），根据州信息表的州名（NAME）**替换为简称（ABBR）**，雇员表文件为emp.txt、州信息表states.txt：

解题思路

使用游标函数cs.run，在原游标上修改字段值。

	A	B
1	=file("emp.txt").cursor@t(EID,NAME,SURNAME,GENDER,STATE,BIRTHDAY)	/创建文件游标，只读指定字段
2	=file("states.txt").import@t()	/读取州信息
3	=A1.run((a=STATE,STATE=A2.select@1(NAME==a).ABBR))	/替换所在州字段为简称
4	=A1.fetch()	/读取原游标修改字段值后的数据

A4

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY
1	1	Rebecca	Moore	F	CA	1974-11-20
2	2	Ashley	Wilson	F	NY	1980-07-19
3	3	Rachel	Johnson	F	NM	1970-12-17
4	4	Emily	Smith	F	TX	1985-03-07
5	5	Ashley	Smith	F	TX	1975-05-13

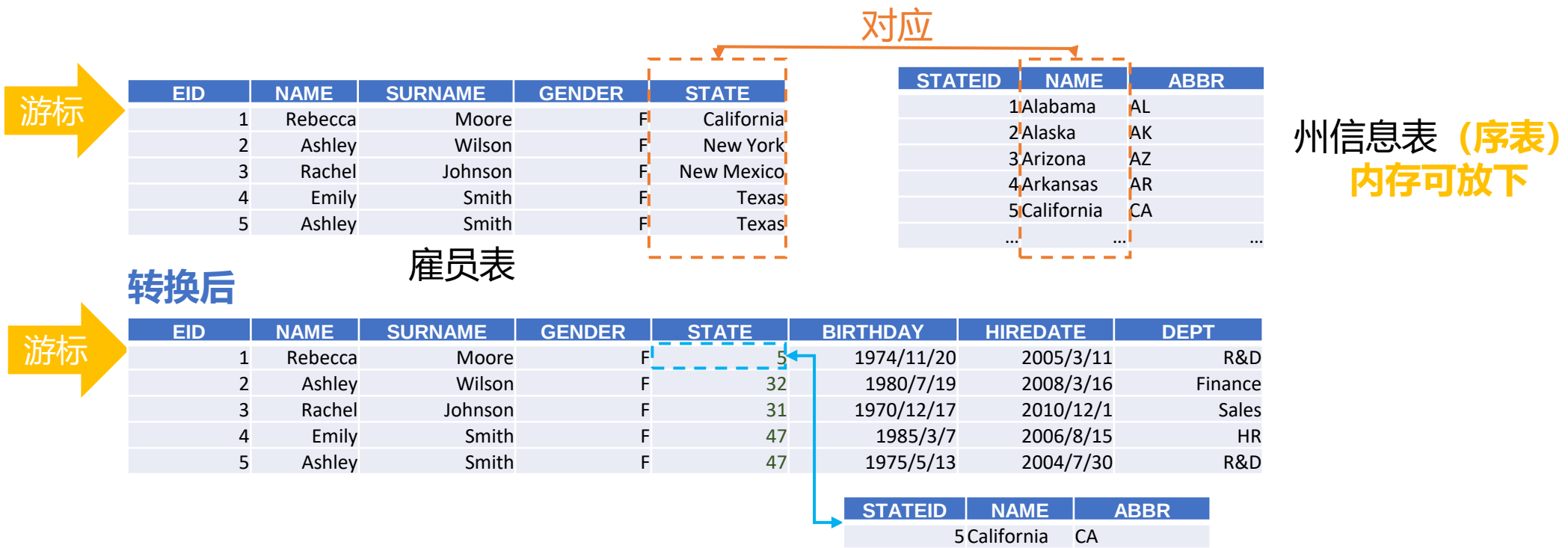
例题使用的数据量较小，fetch出了所有结果，实际运用中，**需要考虑是否会内存溢出**。

结果的前5条记录如图所示。

16.3.2 使用游标函数

外键对象化是处理外键表的常用方法。**第十四章**介绍了数据量较小时使用的函数。数据量较大时，也有类似的游标函数执行**外键对象化（cs.switch）**。注意，**外键表必须是内存放得下的序表。**

例如：将雇员表中所在州（SATAE），**转换**为对应的州记录；



16.3.2 使用游标函数

【例16.13】将雇员表中所在州（SATAE），**转换**为对应的州记录，雇员表文件为emp.txt、州信息表states.txt：

EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT
1	Rebecca	Moore	F	California	1974/11/20	2005/3/11	R&D
2	Ashley	Wilson	F	New York	1980/7/19	2008/3/16	Finance
3	Rachel	Johnson	F	New Mexico	1970/12/17	2010/12/1	Sales
4	Emily	Smith	F	Texas	1985/3/7	2006/8/15	HR
5	Ashley	Smith	F	Texas	1975/5/13	2004/7/30	R&D
6	Matthew	Johnson	M	California	1984/7/7	2005/7/7	Sales
7	Alexis	Smith	F	Illinois	1972/8/16	2002/8/16	Sales
8	Megan	Wilson	F	California	1979/4/19	1984/4/19	Marketing
9	Victoria	Davis	F	Texas	1983/12/7	2009/12/7	HR
10	Ryan	Johnson	M	Pennsylvania	1976/3/12	2006/3/12	R&D
…	…	…	…	…	…	…	…

雇员表（emp.txt）

STATEID	NAME	ABBR
1	Alabama	AL
2	Alaska	AK
3	Arizona	AZ
4	Arkansas	AR
5	California	CA
6	Colorado	CO
7	Connecticut	CT
8	Delaware	DE
9	Florida	FL
10	Georgia	GA
…	…	…

州信息表（states.txt）

16.3.2 使用游标函数

【例16.13】将雇员表中所在州（SATAE），转换为对应的州记录，雇员表文件为emp.txt、州信息表states.txt：

解题思路

使用游标函数cs.switch，在原游标上转换字段值为记录。

	A	B
1	=file("emp.txt").cursor@t(EID,NAME,SURNAME,GENDER,STATE,BIRTHDAY)	/创建文件游标，只读指定字段
2	=file("states.txt").import@t()	/读取州信息
3	=A1.switch(STATE,A2:NAME)	/替换所在州字段为对应记录
4	=A1.fetch()	/读取原游标修改字段值后的数据

A4

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY
1	1	Rebecca	Moore	F	[5,Califor...	1974-11...
2	2	Ashley	Wilson	F	[32,New ...	1980-07...
3	3	Rachel	Johnson	F	[31,New ...	1970-12...
4	4	Emily	Smith	F	[43,Texa...	1985-03...
5	5	Ashley	Smith	F	[43,Texa...	1975-05...

STATEID	NAME	ABBR
5	California	CA

【例16.14】将雇员表里增加新字段保存所在州（SATAE）对应的记录，雇员表文件为emp.txt、州信息表states.txt：

EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT
1	Rebecca	Moore	F	California	1974/11/20	2005/3/11	R&D
2	Ashley	Wilson	F	New York	1980/7/19	2008/3/16	Finance
3	Rachel	Johnson	F	New Mexico	1970/12/17	2010/12/1	Sales
4	Emily	Smith	F	Texas	1985/3/7	2006/8/15	HR
5	Ashley	Smith	F	Texas	1975/5/13	2004/7/30	R&D
6	Matthew	Johnson	M	California	1984/7/7	2005/7/7	Sales
7	Alexis	Smith	F	Illinois	1972/8/16	2002/8/16	Sales
8	Megan	Wilson	F	California	1979/4/19	1984/4/19	Marketing
9	Victoria	Davis	F	Texas	1983/12/7	2009/12/7	HR
10	Ryan	Johnson	M	Pennsylvania	1976/3/12	2006/3/12	R&D
…	…	…	…	…	…	…	…

雇员表（emp.txt）

STATEID	NAME	ABBR
1	Alabama	AL
2	Alaska	AK
3	Arizona	AZ
4	Arkansas	AR
5	California	CA
6	Colorado	CO
7	Connecticut	CT
8	Delaware	DE
9	Florida	FL
…	…	…

州信息表（states.txt）

16.3.2 使用游标函数

【例16.14】将雇员表里增加新字段保存所在州（SATAE）对应的记录，雇员表文件为emp.txt、州信息表states.txt：

解题思路

使用游标函数cs.join，在原游标上增加字段保存外键表记录

	A	B
1	=file("emp.txt").cursor@t(EID,NAME,SURNAME,GENDER,STATE,BIRTHDAY)	/创建文件游标，只读指定字段
2	=file("states.txt").import@t()	/读取州信息
3	=A1.join(STATE,A2:NAME,~:STATER)	/增加新字段保存外键表记录
4	=A1.fetch()	/读取原游标增加字段后的数据

A4

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY	STATER
1	1	Rebecca	Moore	F	California	1974-11-20	2005-03-11	R&D	7000	[5,California,CA]
2	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000	[32,New York,NY]
3	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000	[31,New Mexico,NM]
4	4	Emily	Smith	F	Texas	1985-03-07	2004-07-30	HR	7000	[43,Texas,TX]
5	5	Ashley	Smith	F	Texas	1975-05-13	2005-07-07	R&D	16000	[43,Texas,TX]
…	…	…	…	…	…	…	…	…	…	…

16.3.3 延迟计算与立即计算

立即计算——游标附加的操作会立即执行，比如：cs.groups/cs.total/cs.iterate/cs.id等

例如，订单表（orders.txt）中，统计各年份的订单总数。

	A	B
1	=file("orders.txt").cursor@t()	/打开文件并创建游标
2	=A1.groups(year(O_ORDERDATE):Year;count(~):OrderCount)	/按年份统计订单总数

A2

Index	Year	OrderCount
1	1992	227089
2	1993	226645
3	1994	227597
4	1995	228637
5	1996	228626
6	1997	227783
7	1998	133623

游标的立即计算，是指游标附加操作后，立即遍历游标数据进行计算并返回结果

16.3.3 延迟计算与立即计算

延迟计算——游标附加的操作不会立即计算，而是等到遍历游标时才计算
比如：cs.select/cs.new/cs.derive/cs.run等

例如，查找订单表中客户号（O_CUSTKEY）为4的所有记录

	A	B
1	=file("orders.txt").cursor@t()	/打开文件并创建游标
2	=A1.select(O_CUSTKEY==4)	/从游标中选出客户号为4的订单记录
3	=A2.fetch()	/从游标A2取数

A1

Value
com.raqsoft.dm.cursor.FileCursor@1f48da9f

A1打开文件创建游标，返回游标

A3

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK
164711	4	F	311722.87	1992/4/26	3-MEDIUM	Clerk#000000361
385825	4	O	277493.04	1995/11/1	2-HIGH	Clerk#000000465
1192231	4	O	143971.54	1996/6/3	1-URGENT	Clerk#000000978
1226497	4	F	88317.19	1993/10/4	1-URGENT	Clerk#000000154
...	...	...	...	...	...	...

A3遍历游标时附加的select操作才会得到计算，并返回结果

A2

Value
com.raqsoft.dm.cursor.FileCursor@1f48da9f

A2在A1的游标上附加select操作，
不会立即计算，**仍旧返回游标**

延迟计算的好处在于写起来和内存运算差不多，容易理解，但又不会真地生成中间结果集（内存计算则会）占用空间（或缓存）
立即计算写起来也差不多，但多步骤时时会产生中间结果集占用大量内存

16.4 有序游标

16.4 有序游标

当游标中数据**有序**时，也可以使用类似之前讲过的**序表的有序运算方法**

16.4 有序游标

【例16.15】**订单日期有序**的订单表，按日期分组，取前三组记录，订单表文件为orders.txt：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
3271	33058	F	125535.15	1992/1/1	1-URGENT	Clerk#000000421	0
5607	91021	F	37356.47	1992/1/1	4-NOT SPECIFIED	Clerk#000000137	0
20742	136801	F	121343.42	1992/1/1	4-NOT SPECIFIED	Clerk#000000913	0
23010	91141	F	97372.7	1992/1/1	2-HIGH	Clerk#000000831	0
…	…	…	…	…	…	…	…
3987111	364	F	78703.45	1992/1/2	2-HIGH	Clerk#000000129	0
1141924	700	F	173598.79	1992/1/2	2-HIGH	Clerk#000000603	0
631968	817	F	238394.75	1992/1/2	3-MEDIUM	Clerk#000000939	0
2102785	856	F	173343.08	1992/1/2	5-LOW	Clerk#000000678	0
…	…	…	…	…	…	…	…

16.4 有序游标

【例16.15】订单日期有序的订单表，按日期分组，取前三组记录，订单表文件为orders.txt：

解题思路

用cs.group对有序游标做相邻等值分组

	A	B
1	=file("orders.txt").cursor@t()	/A1打开文件并创建游标
2	=A1.group(O_ORDERDATE)	/按日期做相邻等值分组
3	=A1.fetch@x(3)	/取前三组结果

A3

Index	Member
1	[[3271,33058,F, ...],[5607,91021,F, ...],[20...
2	[[4505602,89161,F, ...],[4512870,18220,F...
3	[[6657,46195,F, ...],[22693,21748,F, ...],[2...

Index	O_ORDER...	O_CUSTKEY	O_ORDER...	O_TOTAL...	O_ORDERDA...
1	3271	33058	F	125535.15	1992-01-01
2	5607	91021	F	37356.47	1992-01-01
3	20742	136801	F	121343.42	1992-01-01
4	23010	91141	F	97372.7	1992-01-01
5	27015	10954	F	41753.82	1992-01-01

16.4 有序游标

【例16.16】**订单日期有序**的订单表，统计每天的订单数，订单表文件为orders.txt：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
3271	33058	F	125535.15	1992/1/1	1-URGENT	Clerk#000000421	0
5607	91021	F	37356.47	1992/1/1	4-NOT SPECIFIED	Clerk#000000137	0
20742	136801	F	121343.42	1992/1/1	4-NOT SPECIFIED	Clerk#000000913	0
23010	91141	F	97372.7	1992/1/1	2-HIGH	Clerk#000000831	0
…	…	…	…	…	…	…	…
3987111	364	F	78703.45	1992/1/2	2-HIGH	Clerk#000000129	0
1141924	700	F	173598.79	1992/1/2	2-HIGH	Clerk#000000603	0
631968	817	F	238394.75	1992/1/2	3-MEDIUM	Clerk#000000939	0
2102785	856	F	173343.08	1992/1/2	5-LOW	Clerk#000000678	0
…	…	…	…	…	…	…	…

16.4 有序游标

【例16.16】**订单日期有序**的订单表，统计每天的订单数，订单表文件为orders.txt：

解题思路

用cs.groups@o对有序游标做相邻等值分组聚合

	A	B
1	=file("orders.txt").cursor@t()	/A1打开文件并创建游标
2	=A1.groups@o(O_ORDERDATE;count(~):count)	/按日期做相邻等值分组聚合

A3

Index	O_ORDERDATE	count
1	1992-01-01	621
2	1992-01-02	612
3	1992-01-03	598
4	1992-01-04	670
5	1992-01-05	641

16.4 有序游标

有时需要对有序游标完成一些比较复杂的计算，无法使用游标函数直接得到结果，这时就需要循环有序游标

游标

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
3271	33058	F	125535.15	1992/1/1	1-URGENT	Clerk#000000421	0
5607	91021	F	37356.47	1992/1/1	4-NOT SPECIFIED	Clerk#000000137	0
20742	136801	F	121343.42	1992/1/1	4-NOT SPECIFIED	Clerk#000000913	0
23010	91141	F	97372.7	1992/1/1	2-HIGH	Clerk#000000831	0
…	…	…	…	…	…	…	…
3987111	364	F	78703.45	1992/1/2	2-HIGH	Clerk#000000129	0
1141924	700	F	173598.79	1992/1/2	2-HIGH	Clerk#000000603	0
631968	817	F	238394.75	1992/1/2	3-MEDIUM	Clerk#000000939	0
2102785	856	F	173343.08	1992/1/2	5-LOW	Clerk#000000678	0
…	…	…	…	…	…	…	…

如：循环订单日期有序的订单表，直至取数条件（O_ORDERDATE）发生变化，统计每天的订单数

1992-01-01的数据

1992-01-02的数据

16.4 有序游标

【例16.17】统计按**订单日期有序**的订单表中的每天的订单数， 订单表文件为orders.txt：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
3271	33058	F	125535.15	1992/1/1	1-URGENT	Clerk#000000421	0
5607	91021	F	37356.47	1992/1/1	4-NOT SPECIFIED	Clerk#000000137	0
20742	136801	F	121343.42	1992/1/1	4-NOT SPECIFIED	Clerk#000000913	0
23010	91141	F	97372.7	1992/1/1	2-HIGH	Clerk#000000831	0
…	…	…	…	…	…	…	…
3987111	364	F	78703.45	1992/1/2	2-HIGH	Clerk#000000129	0
1141924	700	F	173598.79	1992/1/2	2-HIGH	Clerk#000000603	0
631968	817	F	238394.75	1992/1/2	3-MEDIUM	Clerk#000000939	0
2102785	856	F	173343.08	1992/1/2	5-LOW	Clerk#000000678	0
…	…	…	…	…	…	…	…

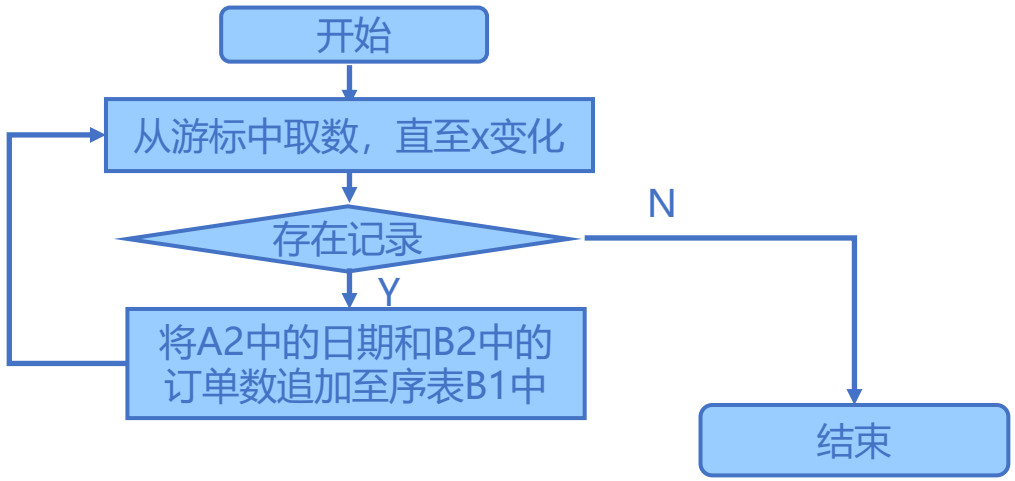
16.4 有序游标

【例16.17】统计按**订单日期有序**的订单表中的每天的订单数，订单表文件为orders.txt：

解题思路

用for cs;x语句来循环游标中的数据，循环读出每次x变化时的数据，按要求进行计算

	A	B	C
1	=file("orders.txt").cursor@t()	=create(Date,Count)	/A1打开文件并创建游标，B1创建空序表
2	for A1;O_ORDERDATE	=A2.count()	/A2按日期循环取数，B2统计当日订单数
3		>B1.insert(0,A2.O_ORDERDATE,B2)	/将A2中的日期和B2中的订单数追加至序表



B1

Date	Count
1992-01-01	621
1992-01-02	612
1992-01-03	598
...	...

16.4 有序游标

【例16.18】查询日志中，最大单次事件记录行数，以及对应的用户标签（usertag）：

```
--flag--
usertag:0ohoi4
pybmbceiodlrafrelayzf
cxpiqlqoftsist
…
mbfcmgocnvixyueky
ogyzbllkwarvt
kjowchyuyxmvdde
--flag--
usertag:5sb9y
evhnjwtvxcy
dbepkgheyeelp
jmqpdrvtfodbzdszlvk
…
```

日志文件（log.txt）
每次事件起始标识为--flag--

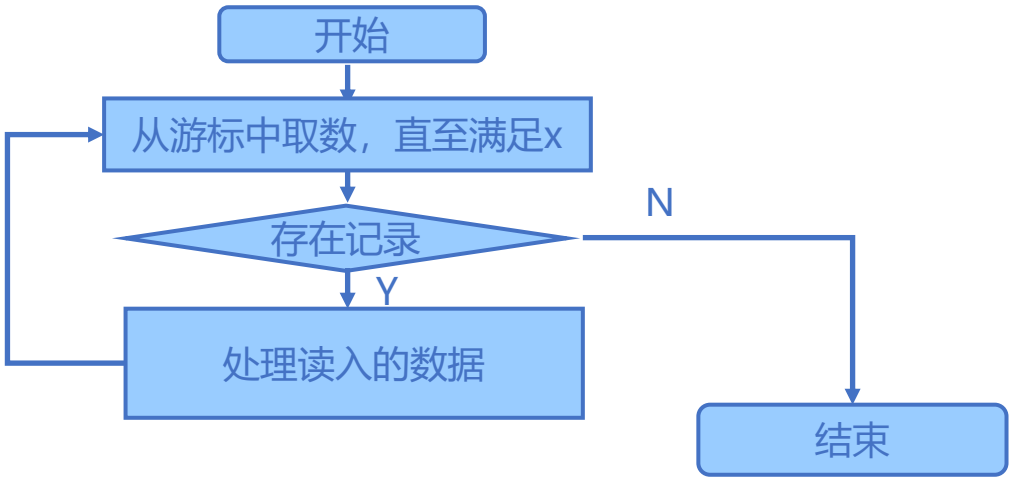
16.4 有序游标

【例16.18】查询日志中，最大单次事件记录行数，以及对应的用户标签（usertag）：

解题思路

用for cs;x语句来循环读取游标中的数据，每次读到符合条件表达式x为真为止

	A	B	C
1	=file("log.txt").cursor@i()		/A1打开文件并创建游标，@i返回序列
2	for A1;~=="--flag--"	=[A2(2),A2.len()-2]	/A1读入单次事件数据，A2获取用户标签和事件记录行数。A2.len()-2去掉头两行
3		>if(B2(2)>B1(2),B1=B2)	/判断当前结果，若满足则替换之前的结果



B1

Index	Member
1	usertag:0ohoi4
2	999998

16.5 大游标

16.5.1 大排序

序表的排序（**详见第十一章——有关表的计算**），介绍了数据量较小，可以放到内存时的排序方法（A.sort）

如果需要排序的数据量巨大，**不能一次将它们读入内存**，普通的排序方法就无法执行了，此时需要使用**外存排序（cs.sortx）**

外存排序时，每次读取**部分数据进行排序**，并将每次排好序的数据写入**临时文件**，最后再对这些临时文件**进行归并排序**，从而获得最终的结果

16.5.1 大排序

【例16.19】从通话记录表中，按拨打者号码进行排序，取出排序后的前100条记录。

通话记录表（records.txt）部分数据如下：

Index	outid	duration
1	18500694558	11
2	18500665932	94
3	18600854809	176
4	18600677317	120
5	18100517144	321
6	18100857113	86
7	18100733553	81
8	18100796252	362
9	18500313020	354
10	18500113597	133
11	18500276539	204
12	18600581342	132
13	18600022612	173
14	18600424740	314
15	18100339412	302

16.5.1 大排序

【例16.19】从通话记录表中，按拨打者号码进行排序，取出排序后的前100条记录。

解题思路

使用cs.sortx函数对游标进行外存排序

	A	B
1	=file("records.txt").cursor@t()	/打开文件并创建游标
2	=A1.sortx(outid)	/使用游标的外存排序函数，按拨打者号码进行排序
3	=A2.fetch(100)	/取前100条
4	>A1.close()	/关闭游标

A3:

Index	outid	duration
1	13600000001	329
2	13600000001	72
3	13600000001	222
4	13600000001	261
5	13600000001	217
6	13600000001	193
7	13600000001	296
8	13600000001	123

16.5.2 大分组

16.3.2一节介绍了**游标的聚合**运算可以使用**cs.groups**函数。但如果聚合后的结果也是**大数据，**就需要借助外存来完成计算。

比如，中小型企业的客户，客户数量可能仅**几万**，按客户分组汇总用全内存模式（**cs.groups**）就足够了。但如果是电信公司，用户数至少在**千万级甚至上亿**，这时分组聚合的结果**内存往往放不下了**，必须得用外存分组（**cs.groupx**）。

大型电商对客户订单情况的统计、电信公司对客户每天上网流量的统计等，都会面临这样的问题。

16.5.2 大分组

【例16.20】从通话记录表中，统计每位用户的拨打电话次数，查看前5条统计结果。

通话记录表（records.txt）部分数据如下：

Index	outid	duration
1	18500694558	11
2	18500665932	94
3	18600854809	176
4	18600677317	120
5	18100517144	321
6	18100857113	86
7	18100733553	81
8	18100796252	362
9	18500313020	354
10	18500113597	133
11	18500276539	204
12	18600581342	132
13	18600022612	173
14	18600424740	314
15	18100339412	302

16.5.2 大分组

【例16.20】从通话记录表中，统计每位用户的拨打电话次数，查看前5条统计结果。

解题思路

使用cs.groupx函数对游标进行外存分组

	A	B
1	=file("records.txt").cursor@t()	/打开文件并创建游标
2	=A1.groupx(outid;count(~):count)	/使用游标的外存分组函数，按拨打者号码进行分组，统计拨打次数
3	=A2.fetch(5)	/取前5条结果
4	>A1.close()	/关闭游标

A3:

Index	outid	count
1	13600000001	27
2	13600000002	34
3	13600000003	27
4	13600000004	23
5	13600000005	19

cs.groupx和cs.sortx.group的结果一样，都是有序的：

	A
1	=file("records.txt").cursor@t()
2	=A1.sortx(outid)
3	=A2.group(outid;count(~):count)
4	=A3.fetch(5)
5	>A1.close()

16.5.3 合并

有时需要处理的数据分布在**多个文件**中，需要**合并**后，再查询或计算。

例如，当不同优先级的订单数据分别在不同的文件中，将各个优先级别的订单记录合并（CS.conjx）。

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	2243/9/13	1992/1/1	1-URGENT
176768	112850	2936/6/26	1992/1/1	1-URGENT
201414	11642	2367/2/8	1992/1/1	1-URGENT
228804	96325	2278/6/10	1992/1/1	1-URGENT
…	…	…	…	…

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
23010	91141	2166/8/4	1992/1/1	2-HIGH
27137	54109	2667/12/17	1992/1/1	2-HIGH
175975	7511	2682/3/11	1992/1/1	2-HIGH
249668	115781	2008/11/5	1992/1/1	2-HIGH
…	…	…	…	…

1-URGENT.txt　　2-HIGH.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	2243/9/13	1992/1/1	1-URGENT
176768	112850	2936/6/26	1992/1/1	1-URGENT
…	…	…	…	…
23010	91141	2166/8/4	1992/1/1	2-HIGH
27137	54109	2667/12/17	1992/1/1	2-HIGH
…	…	…	…	…

16.5.3 合并

【例16.21】将不同优先级的数据文件游标合并，每次取合并后游标中100条记录，若这100条记录中首尾记录的优先级不同，则跳出循环，观察最后取出的100条记录的数据特征

各优先级文件的部分数据如下（**各文件中记录均大于100条**）：

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	125535.00	1992/1/1	1-URGENT
176768	112850	378570.00	1992/1/1	1-URGENT
201414	11642	170608.00	1992/1/1	1-URGENT
228804	96325	138224.00	1992/1/1	1-URGENT
…	…	…	…	…

1-URGENT.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
23010	91141	97372.00	1992/1/1	2-HIGH
27137	54109	280493.00	1992/1/1	2-HIGH
175975	7511	285691.00	1992/1/1	2-HIGH
249668	115781	39757.00	1992/1/1	2-HIGH
…	…	…	…	…

2-HIGH.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
37543	124229	178835.23	1992/1/1	3-MEDIUM
88199	36158	268020.49	1992/1/1	3-MEDIUM
115585	143335	201582.36	1992/1/1	3-MEDIUM
126753	136505	207936.02	1992/1/1	3-MEDIUM
…	…	…	…	…

3-MEDIUM.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
5607	91021	37356.47	1992/1/1	4-NOT SPECIFIED
20742	136801	121343.42	1992/1/1	4-NOT SPECIFIED
45697	109417	345167.47	1992/1/1	4-NOT SPECIFIED
159266	99056	125324.77	1992/1/1	4-NOT SPECIFIED
…	…	…	…	…

4-NOT SPECIFIED.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
27015	10954	41753.82	1992/1/1	5-LOW
59718	24749	83608.04	1992/1/1	5-LOW
139655	25249	118709.27	1992/1/1	5-LOW
190656	23695	154110.00	1992/1/1	5-LOW
…	…	…	…	…

5-LOW.txt

16.5.3 合并

【例16.21】将不同优先级的数据文件游标合并，每次取合并后游标中100条记录，若这100条记录中首尾记录的优先级不同，则跳出循环，观察最后取出的100条记录的数据特征

解题思路

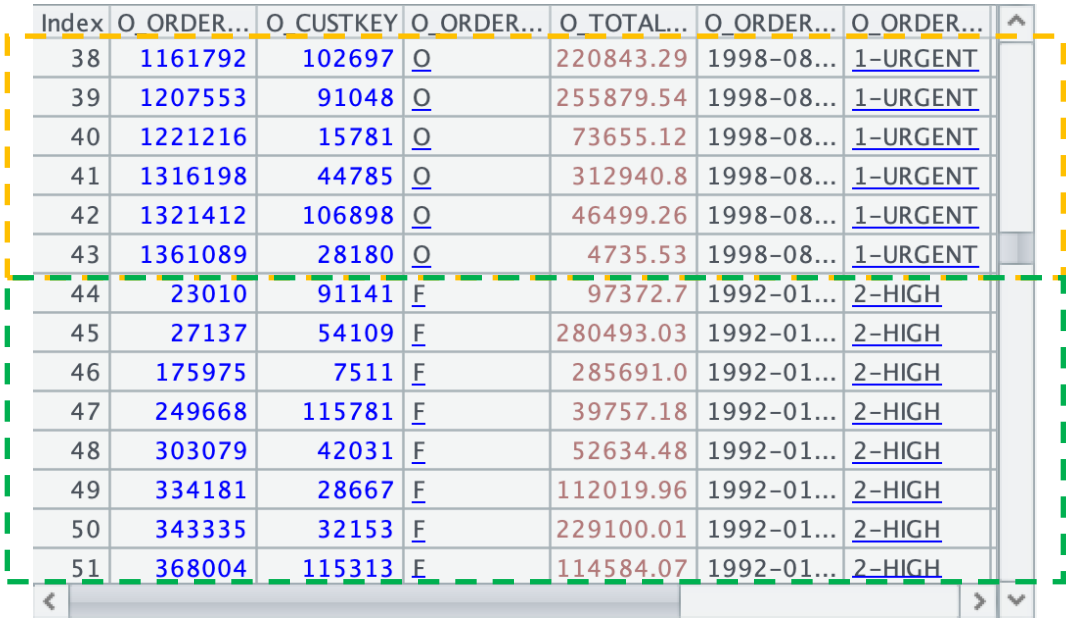
使用CS.conjx函数，合并游标

	A	B	C	D
1	=file("1-URGENT.txt")			/打开文件
2	=file("2-HIGH.txt")			/打开文件
3	=file("3-MEDIUM.txt")			/打开文件
4	=file("4-NOT SPECIFIED.txt")			/打开文件
5	=file("5-LOW.txt")			/打开文件
6	=[A1:A5].(~.cursor@t())			/对文件序列的每个文件建立游标
7	=A6.conjx()	100		/将游标序列合并
8	for	=A7.fetch(B7)		/循环游标中的数据，每次取100条
9		=B8(1).O_ORDERPRIORITY	=B8.m(-1).O_ORDERPRIORITY	/观察首条和末尾记录的优先级
10		if B9!=C9	break	/若优先级不一致，则跳出循环
11	>A7.close()			/关闭游标

16.5.3 合并

【例16.21】将不同优先级的数据文件游标合并，每次取合并后游标中100条记录，若这100条记录中首尾记录的优先级不同，则跳出循环，观察最后取出的100条记录的数据特征

执行完上一页的代码，观察B8中的结果，如下：



Index	O_ORDER...	O_CUSTKEY	O_ORDER...	O_TOTAL...	O_ORDER...	O_ORDER...
38	1161792	102697	O	220843.29	1998-08...	1-URGENT
39	1207553	91048	O	255879.54	1998-08...	1-URGENT
40	1221216	15781	O	73655.12	1998-08...	1-URGENT
41	1316198	44785	O	312940.8	1998-08...	1-URGENT
42	1321412	106898	O	46499.26	1998-08...	1-URGENT
43	1361089	28180	O	4735.53	1998-08...	1-URGENT
44	23010	91141	F	97372.7	1992-01...	2-HIGH
45	27137	54109	F	280493.03	1992-01...	2-HIGH
46	175975	7511	F	285691.0	1992-01...	2-HIGH
47	249668	115781	F	39757.18	1992-01...	2-HIGH
48	303079	42031	F	52634.48	1992-01...	2-HIGH
49	334181	28667	F	112019.96	1992-01...	2-HIGH
50	343335	32153	F	229100.01	1992-01...	2-HIGH
51	368004	115313	F	114584.07	1992-01...	2-HIGH

可见合并后的游标，会根据原游标序列的顺序合并数据

16.5.4 归并

有时需要处理的数据分布在**多个文件**中，并且都对某些字段**有序**，使用合并运算会导致总结果无序，而使用**归并**运算，则可以保持原来的次序，以便进一步查询或计算。

例如，不同优先级的订单数据分别在不同的文件中（按日期有序），将不同优先级的数据文件按日期进行有序归并（**CS.megerx**）。

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	2243/9/13	1992/1/1	1-URGENT
176768	112850	2936/6/26	1992/1/1	1-URGENT
201414	11642	2367/2/8	1992/1/1	1-URGENT
228804	96325	2278/6/10	1992/1/1	1-URGENT
…	…	…	…	…

1-URGENT.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
175975	7511	2682/3/11	1992/1/1	2-HIGH
…	…	…	…	…
23010	91141	2166/8/4	1992/1/2	2-HIGH
27137	54109	2667/12/17	1992/1/2	2-HIGH
…	…	…	…	…

2-HIGH.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	2243/9/13	1992/1/1	1-URGENT
176768	112850	2936/6/26	1992/1/1	1-URGENT
…	…	…	…	…
23010	91141	2166/8/4	1992/1/2	2-HIGH
27137	54109	2667/12/17	1992/1/2	2-HIGH
…	…	…	…	…

16.5.4 归并

【例16.22】将不同优先级的数据文件按日期进行**有序归并**，取前200条记录观察数据特征

各优先级文件的部分数据如下（**各文件中对O_ORDERDATE有序且每天的数据小于200条**）：

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	125535.00	1992/1/1	1-URGENT
176768	112850	378570.00	1992/1/1	1-URGENT
201414	11642	170608.00	1992/1/1	1-URGENT
228804	96325	138224.00	1992/1/1	1-URGENT
…	…	…	…	…

1-URGENT.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
23010	91141	97372.00	1992/1/1	2-HIGH
27137	54109	280493.00	1992/1/1	2-HIGH
175975	7511	285691.00	1992/1/1	2-HIGH
249668	115781	39757.00	1992/1/1	2-HIGH
…	…	…	…	…

2-HIGH.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
37543	124229	178835.23	1992/1/1	3-MEDIUM
88199	36158	268020.49	1992/1/1	3-MEDIUM
115585	143335	201582.36	1992/1/1	3-MEDIUM
126753	136505	207936.02	1992/1/1	3-MEDIUM
…	…	…	…	…

3-MEDIUM.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
5607	91021	37356.47	1992/1/1	4-NOT SPECIFIED
20742	136801	121343.42	1992/1/1	4-NOT SPECIFIED
45697	109417	345167.47	1992/1/1	4-NOT SPECIFIED
159266	99056	125324.77	1992/1/1	4-NOT SPECIFIED
…	…	…	…	…

4-NOT SPECIFIED.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
27015	10954	41753.82	1992/1/1	5-LOW
59718	24749	83608.04	1992/1/1	5-LOW
139655	25249	118709.27	1992/1/1	5-LOW
190656	23695	154110.00	1992/1/1	5-LOW
…	…	…	…	…

5-LOW.txt

16.5.4 归并

【例16.22】将不同优先级的数据文件按日期进行**有序归并**，取前200条记录观察数据特征

解题思路

使用CS.mergex(x)函数，有序归并。有序归并是将多个有序数据合并成一个有序数据的方法。

	A	B
1	=file("1-URGENT.txt")	/打开文件
2	=file("2-HIGH.txt")	/打开文件
3	=file("3-MEDIUM.txt")	/打开文件
4	=file("4-NOT SPECIFIED.txt")	/打开文件
5	=file("5-LOW.txt")	/打开文件
6	=[A1:A5].(~.cursor@t())	/对文件序列的每个文件建立游标
7	=A6.mergex(O_ORDERDATE)	/将游标序列按日期进行有序归并
8	=A7.fetch(200)	/取200条记录
9	>A7.close()	/关闭游标

16.5.4 归并

【例16.22】将不同优先级的数据文件按日期进行**有序归并**，取前200条记录观察数据特征

执行完上一页的代码，观察A8中的结果，如下：

Index	O_ORDER...	O_CUSTKEY	O_ORDER...	O_TOTAL...	O_ORDERDATE	O_ORDER...
116	5586306	147610	F	80531.02	1992-01-01	1-URGENT
117	5660774	53731	F	107817.53	1992-01-01	1-URGENT
118	5714757	58784	F	144932.69	1992-01-01	1-URGENT
119	5772962	55256	F	172489.4	1992-01-01	1-URGENT
120	5854464	41117	F	246673.86	1992-01-01	1-URGENT
121	5885633	71485	F	191812.25	1992-01-01	1-URGENT
122	5998247	14444	F	210703.01	1992-01-01	1-URGENT
123	23010	91141	F	97372.7	1992-01-01	2-HIGH
124	27137	54109	F	280493.03	1992-01-01	2-HIGH
125	175975	7511	F	285691.0	1992-01-01	2-HIGH
126	249668	115781	F	39757.18	1992-01-01	2-HIGH
127	303079	42031	F	52634.48	1992-01-01	2-HIGH
128	334181	28667	F	112019.96	1992-01-01	2-HIGH
129	343335	32153	F	229100.01	1992-01-01	2-HIGH

可见按日期有序归并的数据并没有像**合并（CS.conjx）**那样按原游标序列的顺序获取数据，而是按照订单的日期有序来获取不同优先级的数据。

使用有序**归并（CS.mergex(x)）**时， 游标序列CS中的每个游标都必须均对**表达式x有序**。

16.5.5 连接

有时需要综合多个游标中有关联的数据，合起来生成一张宽表。
这时候要求各个游标对关联字段有序

例如，不同优先级的订单数据分别存在按**日期有序**的文件中，统计每天各优先级别的订单记录数，这需要将不同优先级的数据按日期**连接（joinx）**。

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	125535.00	1992/1/1	1-URGENT
176768	112850	378570.00	1992/1/1	1-URGENT
201414	11642	170608.00	1992/1/1	1-URGENT
228804	96325	138224.00	1992/1/1	1-URGENT
…	…	…	…	…

1-URGENT.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
23010	91141	97372.00	1992/1/1	2-HIGH
27137	54109	280493.00	1992/1/1	2-HIGH
175975	7511	285691.00	1992/1/1	2-HIGH
249668	115781	39757.00	1992/1/1	2-HIGH
…	…	…	…	…

2-HIGH.txt

O_ORDERPRIORITY	O_ORDERDATE	COUNT	O_ORDERPRIORITY	O_ORDERDATE	COUNT
1-URGENT	1992/1/1	283	2-HIGH	1992/1/1	315
1-URGENT	1992/1/2	749	2-HIGH	1992/1/2	246
…	…	…	…	…	…

16.5.5 连接

【例16.23】统计每天各优先级别的订单记录数。

各优先级文件的部分数据如下：

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	125535.00	1992/1/1	1-URGENT
176768	112850	378570.00	1992/1/1	1-URGENT
201414	11642	170608.00	1992/1/1	1-URGENT
228804	96325	138224.00	1992/1/1	1-URGENT
…	…	…	…	…

1-URGENT.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
23010	91141	97372.00	1992/1/1	2-HIGH
27137	54109	280493.00	1992/1/1	2-HIGH
175975	7511	285691.00	1992/1/1	2-HIGH
249668	115781	39757.00	1992/1/1	2-HIGH
…	…	…	…	…

2-HIGH.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
37543	124229	178835.23	1992/1/1	3-MEDIUM
88199	36158	268020.49	1992/1/1	3-MEDIUM
115585	143335	201582.36	1992/1/1	3-MEDIUM
126753	136505	207936.02	1992/1/1	3-MEDIUM
…	…	…	…	…

3-MEDIUM.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
5607	91021	37356.47	1992/1/1	4-NOT SPECIFIED
20742	136801	121343.42	1992/1/1	4-NOT SPECIFIED
45697	109417	345167.47	1992/1/1	4-NOT SPECIFIED
159266	99056	125324.77	1992/1/1	4-NOT SPECIFIED
…	…	…	…	…

4-NOT SPECIFIED.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
27015	10954	41753.82	1992/1/1	5-LOW
59718	24749	83608.04	1992/1/1	5-LOW
139655	25249	118709.27	1992/1/1	5-LOW
190656	23695	154110.00	1992/1/1	5-LOW
…	…	…	…	…

5-LOW.txt

16.5.5 连接

【例16.23】统计每天各优先级别的订单记录数。

解题思路

用joinx函数对多个同序游标进行连接

	A	B
1	=file("1-URGENT.txt").cursor@t()	/打开文件并创建游标
2	=file("2-HIGH.txt").cursor@t()	/打开文件并创建游标
3	=file("3-MEDIUM.txt").cursor@t()	/打开文件并创建游标
4	=file("4-NOT SPECIFIED.txt").cursor@t()	/打开文件并创建游标
5	=file("5-LOW.txt").cursor@t()	/打开文件并创建游标
6	=A1.groupx(O_ORDERDATE,O_ORDERPRIORITY;count(~):count)	/按订单日期、优先级分组统计订单数
7	=A2.groupx(O_ORDERDATE,O_ORDERPRIORITY;count(~):count)	/按订单日期、优先级分组统计订单数
8	=A3.groupx(O_ORDERDATE,O_ORDERPRIORITY;count(~):count)	/按订单日期、优先级分组统计订单数
9	=A4.groupx(O_ORDERDATE,O_ORDERPRIORITY;count(~):count)	/按订单日期、优先级分组统计订单数
10	=A5.groupx(O_ORDERDATE,O_ORDERPRIORITY;count(~):count)	/按订单日期、优先级分组统计订单数
11	=joinx@1(A6:URGENT,O_ORDERDATE;A7:HIGH,O_ORDERDATE;A8:MEDIUM,O_ORDERDATE;A9:NOT_SPECIFIED,O_ORDERDATE;A10:LOW,O_ORDERDATE)	/对各优先级的聚合结果，按订单日期连接
12	=A11.fetch(10)	/取10条
13	>A11.close()	/关闭游标

16.5.5 连接

【例16.23】统计每天各优先级别的订单记录数。

执行完上一页的代码，观察A12中的结果，如下：

Index	URGENT	HIGH	MEDIUM	NOT_SPECIFIED	LOW
1	[1992-01-01,...	[1992-01-01,...	[1992-01-01,...	[1992-01-01,...	[1992-01-01,...
2	[1992-01-02,...	[1992-01-02,...	[1992-01-02,...	[1992-01-02,...	[1992-01-02,...
3	[1992-01-03,...	[1992-01-03,...	[1992-01-03,...	[1992-01-03,...	[1992-01-03,...
4	[1992-01-04,...	[1992-01-04,...	[1992-01-04,...	[1992-01-04,...	[1992-01-04,...
5	[1992-01-05,...	[1992-01-05,...	[1992-01-05,...	[1992-01-05,...	[1992-01-05,...
6	[1992-01-06,...	[1992-01-06,...	[1992-01-06,...	[1992-01-06,...	[1992-01-06,...
7	[1992-01-07,...	[1992-01-07,...	[1992-01-07,...	[1992-01-07,...	[1992-01-07,...
8	[1992-01-08,...	[1992-01-08,...	[1992-01-08,...	[1992-01-08,...	[1992-01-08,...
9	[1992-01-09,...	[1992-01-09,...	[1992-01-09,...	[1992-01-09,...	[1992-01-09,...
10	[1992-01-10,...	[1992-01-10,...	[1992-01-10,...	[1992-01-10,...	[1992-01-10,...

O_ORDERDATE	O_ORDERPRIORITY	count
1992-01-01	4-NOT SPECIFIED	126

O_ORDERDATE	O_ORDERPRIORITY	count
1992-01-01	5-LOW	115

16.5.5 连接

最终结果若需要返回为序表，例如返回日期、每个优先级下的订单个数，可以在A11后使用new

	A	B
…	…	…
11	=joinx@1(A6:URGENT,O_ORDERDATE;A7:HIGH,O_ORDERDATE;A8:MEDIUM,O_ORDERDATE;A9:NOT_SPECIFIED,O_ORDERDATE;A10:LOW,O_ORDERDATE)	/对各优先级的聚合结果，按订单日期做对齐式连接
12	=A11.new(URGENT.O_ORDERDATE:ORDERDATE,URGENT.count:URGENT,HIGH.count:HIGH,MEDIUM.count:MEDIUM,NOT_SPECIFIED.count:NOT_SPECIFIED,LOW.count:LOW)	/游标附加new运算，结构变为日期及每个优先级的订单数
13	=A12.fetch(10)	/取10条
14	>A12.close()	/关闭游标

Index	ORDERDATE	URGENT	HIGH	MEDIUM	NOT_SPE...	LOW
1	1992-01-01	122	120	138	126	115
2	1992-01-02	113	126	125	119	129
3	1992-01-03	132	119	113	127	107
4	1992-01-04	103	144	133	149	141
5	1992-01-05	146	118	130	129	118
6	1992-01-06	116	128	127	132	123
7	1992-01-07	124	136	109	128	142
8	1992-01-08	120	124	124	121	117
9	1992-01-09	127	112	129	136	138
10	1992-01-10	138	135	114	121	118

16.5.6 多路游标

处理大数据文件时，还可以利用**多核CPU**，对文件**分段并行计算**

如：查找订单表中交易金额（O_TOTALPRICE）大于20万的订单（**订单表按日期O_ORDERDATE有序**）

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE
1	9527	O	173665.47	1992/1/2
2	4	O	46929.18	1992/12/1
...	...	...	...	...
24903795	2732	F	823642.87	1993/6/25

第1段

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE
24903796	62017	O	225029.87	1993/6/28
24903797	14073	F	87784.83	1993/7/30
...	...	...	...	...
49971212	37621	O	62537.17	1994/9/30

第2段

4路游标
各段数据的部
分内容

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE
49971213	82265	F	167087.15	1995/5/6
49971214	38273	O	142775.84	1995/12/15
...	...	...	...	...
73564722	12653	F	72638.21	

第3段

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE
73564723	36901	O	173665.47	1996/1/2
73564724	23126	O	46929.18	1996/12/1
73564725	22673	O	82652.39	1996/12/1
...	...	...	...	...

第4段

16.5.6 多路游标

【例16.24】查找订单表中交易金额大于20万的订单记录有多少个（创建4路游标，完成计算）

订单表文件为orders.txt：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	4	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
…	…	…	…	…	…	…	…
10001	62017	O	225029.87	1998/6/28	4-NOT SPECIFIED	Clerk#000000177	0
10002	114073	F	87784.83	1995/1/30	2-HIGH	Clerk#000000498	0
…	…	…	…	…	…	…	…
20001	4	F	167087.15	1992/5/6	4-NOT SPECIFIED	Clerk#000000121	0
20002	60173	O	142775.84	1995/12/15	5-LOW	Clerk#000000268	0
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…

16.5.6 多路游标

【例16.24】查找订单表中交易金额大于20万的订单记录有多少个（创建4路游标，完成计算）

解题思路

使用游标函数的选项@m，生成多路游标

	A	B
1	=file("orders.txt").cursor@tm(;4)	/打开文件，创建4路游标
2	=A1.select(O_TOTALPRICE>200000).total(count(~))	/对多路游标附加过滤函数，并统计个数

A3:

Value
450996

16.5.6 多路游标

有时数据可能来自于多个游标，这种情况也可以使用**多路游标（CS.mcursor）**，采用并行的方式执行计算。

例如，当不同优先级的订单数据分别在不同的文件中，查找订购日期为1992年3月1号的所有优先级的订单。

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	125535.00	1992/1/1	1-URGENT
176768	112850	378570.00	1992/1/1	1-URGENT
…	…	…	…	…
153271	33058	178835.23	1992/3/1	1-URGENT
196768	112850	268020.49	1992/3/1	1-URGENT
…	…	…	…	…

线程1：查找URGENT.txt文件

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
23010	91141	97372.00	1992/1/1	2-HIGH
27137	54109	280493.00	1992/1/1	2-HIGH
…	…	…	…	…
223010	91141	207936.02	1992/3/1	2-HIGH
437137	54109	178835.23	1992/3/1	2-HIGH
…	…	…	…	…

线程2：查找HIGH.txt文件

合并结果：

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
153271	33058	178835.23	1992/3/1	1-URGENT
196768	112850	268020.49	1992/3/1	1-URGENT
…	…	…	…	…
223010	91141	207936.02	1992/3/1	2-HIGH
437137	54109	178835.23	1992/3/1	2-HIGH
…	…	…	…	…

16.5.6 多路游标

【例16.25】查找订购日期为1992年3月1号的所有优先级的订单，最终结果按O_ORDERKEY有序

各优先级文件的部分数据如下（**各文件按O_ORDERKEY有序**）：

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	125535.00	1992/1/1	1-URGENT
176768	112850	378570.00	1992/1/1	1-URGENT
201414	11642	170608.00	1992/1/1	1-URGENT
228804	96325	138224.00	1992/1/1	1-URGENT
…	…	…	…	…

1-URGENT.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
23010	91141	97372.00	1992/1/1	2-HIGH
27137	54109	280493.00	1992/1/1	2-HIGH
175975	7511	285691.00	1992/1/1	2-HIGH
249668	115781	39757.00	1992/1/1	2-HIGH
…	…	…	…	…

2-HIGH.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
37543	124229	178835.23	1992/1/1	3-MEDIUM
88199	36158	268020.49	1992/1/1	3-MEDIUM
115585	143335	201582.36	1992/1/1	3-MEDIUM
126753	136505	207936.02	1992/1/1	3-MEDIUM
…	…	…	…	…

3-MEDIUM.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
5607	91021	37356.47	1992/1/1	4-NOT SPECIFIED
20742	136801	121343.42	1992/1/1	4-NOT SPECIFIED
45697	109417	345167.47	1992/1/1	4-NOT SPECIFIED
159266	99056	125324.77	1992/1/1	4-NOT SPECIFIED
…	…	…	…	…

4-NOT SPECIFIED.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
27015	10954	41753.82	1992/1/1	5-LOW
59718	24749	83608.04	1992/1/1	5-LOW
139655	25249	118709.27	1992/1/1	5-LOW
190656	23695	154110.00	1992/1/1	5-LOW
…	…	…	…	…

5-LOW.txt

16.5.6 多路游标

【例16.25】查找订购日期为1992年3月1号的所有优先级的订单，最终结果按O_ORDERKEY有序

解题思路

使用CS.mcursor函数，将多游标形成多路游标，多路游标在各个游标同序时按同序字段mergex时也是有序的

	A	B
1	=file("1-URGENT.txt")	/打开文件
2	=file("2-HIGH.txt")	/打开文件
3	=file("3-MEDIUM.txt")	/打开文件
4	=file("4-NOT SPECIFIED.txt")	/打开文件
5	=file("5-LOW.txt")	/打开文件
6	=[A1:A5].(~.cursor@t()).mcursor()	/用A1~A5中的文件生成游标序列，再创建多路游标
7	=A6.select(O_ORDERDATE==date("1992-03-01")).mergex(O_ORDERKEY).fetch()	/过滤订购日期为1992-03-01的数据，按订单号归并后取出结果

A7:

Index	O_ORDER...	O_CUSTKEY	O_...	O_TOTAL...	O_ORDERD...	O_ORDER...	O_CLERK
1	16740	27046	F	101843.93	1992-03-01	5-LOW	Clerk#00...
2	45219	100177	F	246351.96	1992-03-01	5-LOW	Clerk#00...
3	56292	119560	F	304043.03	1992-03-01	5-LOW	Clerk#00...
4	59204	71213	F	250535.19	1992-03-01	3-MEDIUM	Clerk#00...
5	61091	36157	F	9558.5	1992-03-01	3-MEDIUM	Clerk#00...
6	78720	102899	F	191306.41	1992-03-01	2-HIGH	Clerk#00...
7	81057	129427	F	169503.95	1992-03-01	4-NOT SP...	Clerk#00...

16.6 习题

16.6 习题

【习题16.1】读取订单表（orders.txt）中的前两条记录，跳过三条记录后，再读取四条记录。

订单表（orders.txt）前10条数据如下：

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	78002	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0
3	123314	F	193846.25	1993/10/14	5-LOW	Clerk#000000955	0
4	136777	O	32151.78	1995/10/11	5-LOW	Clerk#000000124	0
5	44485	F	144659.2	1994/7/30	5-LOW	Clerk#000000925	0
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED	Clerk#000000058	0
7	39136	O	252004.18	1996/1/10	2-HIGH	Clerk#000000470	0
32	130057	O	208660.75	1995/7/16	2-HIGH	Clerk#000000616	0
33	66958	F	163243.98	1993/10/27	3-MEDIUM	Clerk#000000409	0
...	...	...	...	...	...	...	...

16.6 习题

【习题16.1】读取订单表（orders.txt）中的前两条记录，跳过三条记录后，再读取四条记录。

解题思路

使用cs.fetch(n)函数读取游标中的记录，使用cs.skip(n)函数将游标跳过指定记录数

	A	B
1	=file("orders.txt")	/打开文件名为orders.txt的文件
2	=A1.cursor@t()	/对文件创建游标
3	=A2.fetch(2)	/从游标中读出2条记录
4	=A2.skip(3)	/游标跳过3条记录
5	=A2.fetch(4)	/继续从游标中读出4条记录
6	>A2.close()	/游标数据并未全部取出，关闭游标

仅读取所需要的列时，A2可以改为：

	A	B
2	=A1.cursor@t(O_ORDERKEY, O_CUSTKEY, O_TOTALPRICE)	/创建游标，只读取指定字段

执行结果：

A3:

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
1	36901	O	173665.47	1996/1/2	5-LOW	Clerk#000000951	0
2	78002	O	46929.18	1996/12/1	1-URGENT	Clerk#000000880	0

A5:

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
6	55624	F	58749.59	1992/2/21	4-NOT SPECIFIED	Clerk#000000058	0
7	39136	O	252004.18	1996/1/10	2-HIGH	Clerk#000000470	0
32	130057	O	208660.75	1995/7/16	2-HIGH	Clerk#000000616	0
33	66958	F	163243.98	1993/10/27	3-MEDIUM	Clerk#000000409	0

16.6 习题

【习题16.2】查找通话记录表（record.txt）中拨打者号码（OUTID）为13781722635的通话记录。

通话记录表（record.txt）部分数据如下：

SERIALNUMBER	OUTID	INID	CALLTIME	DURATION	CHARGE
1	13781722635	18635273625	2006-08-11 08:53:22	97	0.00
2	18572635487	18173625183	2006-08-11 09:01:03	108	0.15
3	18572635182	13526384957	2006-08-11 09:02:11	34	0.00
4	13782769374	18502938475	2006-08-11 09:24:21	87	0.00
5	13672394817	18591820385	2006-08-11 09:33:12	132	0.15
6	18638274625	18108273645	2006-08-11 09:35:37	876	0.00
7	18127364918	13582937465	2006-08-11 09:43:18	56	0.15
…	…	…	…	…	…

16.6 习题

【习题16.2】查找通话记录表（record.txt）中拨打者号码（OUTID）为13781722635的通话记录。

解题思路

使用游标函数cs.select查找相应记录

	A	B
1	=file("record.txt").cursor@t()	/打开文件并创建游标
2	=A1.select(OUTID==13781722635)	/从游标中选出拨打者号为13781722635的通话记录
3	=A2.fetch()	/从游标A2取数

A1

Value
com.raqsoft.dm.cursor.FileCursor@1f48da9f

A2

Value
com.raqsoft.dm.cursor.FileCursor@1f48da9f

A3

SERIALNUMBER	OUTID	INID	CALLTIME	DURATION	CHARGE
1	13781722635	18635273625	2006-08-11 08:53:22	97	0.00
273	13781722635	18173625183	2006-08-15 09:01:03	108	0.15
3715	13781722635	13526384957	2006-08-18 09:02:11	34	0.00
7625	13781722635	18502938475	2006-08-23 09:24:21	87	0.00
…	…	…	…	…	…

A2中的结果和A1是相同的。在游标中使用select函数时，返回的结果是其本身，计算A2时其实并未真正从游标中获取数据，而是在游标上附加操作。在A3中执行fetch时，才会根据条件在A1游标中检索数据。

16.6 习题

【习题16.3】统计通话记录表中每天的通话次数，通话记录表文件为record.txt：

SERIALNUMBER	OUTID	INID	CALLTIME	DURATION	CHARGE
1	13781722635	18635273625	2006-08-11 08:53:22	97	0.00
2	18572635487	18173625183	2006-08-11 09:01:03	108	0.15
3	18572635182	13526384957	2006-08-11 09:02:11	34	0.00
4	13782769374	18502938475	2006-08-11 09:24:21	87	0.00
5	13672394817	18591820385	2006-08-11 09:33:12	132	0.15
6	18638274625	18108273645	2006-08-11 09:35:37	876	0.00
7	18127364918	13582937465	2006-08-11 09:43:18	56	0.15
…	…	…	…	…	…

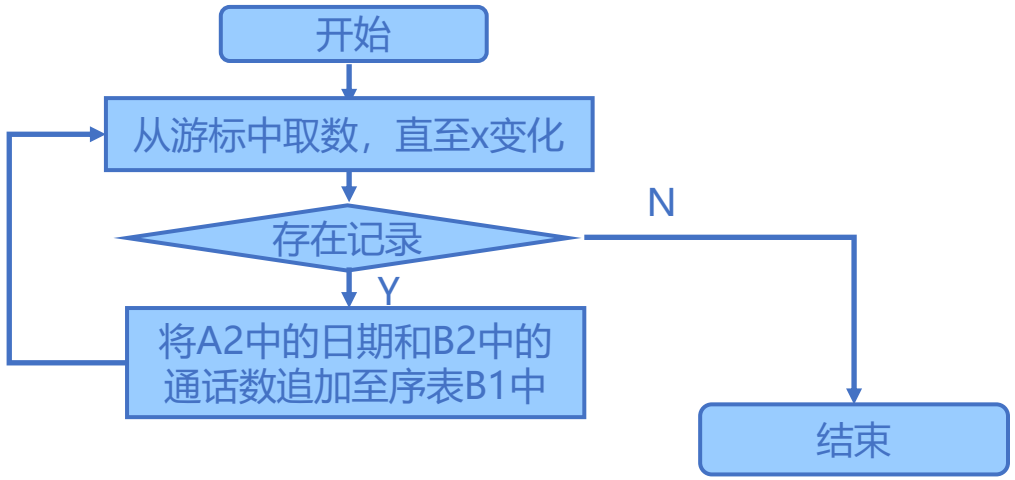
16.6 习题

【习题16.3】统计通话记录表中每天的通话次数，通话记录表文件为record.txt：

解题思路

用for cs;x语句来循环游标中的数据，循环读出每次x变化时的数据，按要求进行计算

	A	B	C
1	=file("record.txt").cursor@t().run(CALLTIME=date(CALLTIME))	=create(Date,Count)	/A1打开文件并创建游标，将CALLTIME变为date类型，B1创建空序表
2	for A1;CALLTIME	=A2.count()	/A2按日期循环取数，B2统计当日通话数
3		>B1.insert(0,A2.CALLTIME,B2)	/将A2中的日期和B2中的通话数追加至序表



B1

Date	Count
2006-08-11	621
2006-08-12	612
2006-08-13	598
...	...

16.6 习题

【习题16.4】统计通话记录表中，每位用户通话次数、平均通话时长和总通话费用

通话记录表文件为record.txt：

SERIALNUMBER	OUTID	INID	CALLTIME	DURATION	CHARGE
1	13781722635	18635273625	2006-08-11 08:53:22	97	0.00
2	18572635487	18173625183	2006-08-11 09:01:03	108	0.15
3	18572635182	13526384957	2006-08-11 09:02:11	34	0.00
4	13782769374	18502938475	2006-08-11 09:24:21	87	0.00
5	13672394817	18591820385	2006-08-11 09:33:12	132	0.15
6	18638274625	18108273645	2006-08-11 09:35:37	876	0.00
7	18127364918	13582937465	2006-08-11 09:43:18	56	0.15
…	…	…	…	…	…

16.6 习题

【习题16.4】统计通话记录表中，每位用户通话次数、平均通话时长和总通话费用

解题思路

使用cs.groups函数，完成各类聚合运算

	A	B
1	=file("record.txt").cursor@t()	/打开文件并创建游标
2	=A1.groups(OUTID;count(~):Count,avg(DURATION):AvgDuration,sum(CHARGE):Total)	/按每位用户统计通话次数、平均通话时长和总通话费用

A2：

OUTID	Count	AvgDuration	Total
13672394817	3	364	4.15
13782769374	7	672	9.55
13782769482	1	87	0.15
13782769674	9	1024	5.15
18127364983	6	781	3.45
18127364996	17	2038	9.95
18127365109	33	4095	13.45
…	…	…	…

16.6 习题

【习题16.5】雇员表（emp.txt）和州信息表（states.txt）

要求：1、将雇员表中的名字（NAME）和姓氏（SURNAME）合并为全名（FullName）；

2、将雇员表中所在州（SATAE），根据州信息表的州名（NAME）替换为简称（ABBR）；

3、根据雇员生日（BIRTHDAY），添加计算列：年龄（Age）。

4、取前十条记录

两表的部分数据如下：

EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT
1	Rebecca	Moore	F	California	1974/11/20	2005/3/11	R&D
2	Ashley	Wilson	F	New York	1980/7/19	2008/3/16	Finance
3	Rachel	Johnson	F	New Mexico	1970/12/17	2010/12/1	Sales
4	Emily	Smith	F	Texas	1985/3/7	2006/8/15	HR
5	Ashley	Smith	F	Texas	1975/5/13	2004/7/30	R&D
6	Matthew	Johnson	M	California	1984/7/7	2005/7/7	Sales
7	Alexis	Smith	F	Illinois	1972/8/16	2002/8/16	Sales
8	Megan	Wilson	F	California	1979/4/19	1984/4/19	Marketing
9	Victoria	Davis	F	Texas	1983/12/7	2009/12/7	HR
10	Ryan	Johnson	M	Pennsylvania	1976/3/12	2006/3/12	R&D
…	…	…	…	…	…	…	…

雇员表（emp.txt）

STATEID	NAME	ABBR
1	Alabama	AL
2	Alaska	AK
3	Arizona	AZ
4	Arkansas	AR
5	California	CA
6	Colorado	CO
7	Connecticut	CT
8	Delaware	DE
9	Florida	FL
10	Georgia	GA
…	…	…

州信息表（states.txt）

16.6 习题

解题思路

可以用游标函数cs.new完成

	A	B
1	=file("emp.txt").cursor@t(EID,NAME,SURNAME,GENDER,STATE,BIRTHDAY)	/对文件选出指定字段创建游标
2	=file("states.txt").import@t()	/读取州信息
3	=A1.new(EID,NAME+" "+SURNAME:FullName,GENDER,A2.select@1(NAME==STATE).ABBR:STATE,BIRTHDAY,age(BIRTHDAY):Age)	/将名字与姓氏合并为全名，替换所在州字段为简称，通过生日计算年龄Age
4	=A1.fetch(10)	/查看前10条数据
5	>A1.close()	/游标数据未全读完，主动关闭游标

A4：

Index	EID	FullName	GENDER	STATE	BIRTHDAY	Age
1	1	Rebecca Moore	F	CA	1974-11-20	45
2	2	Ashley Wilson	F	NY	1980-07-19	39
3	3	Rachel Johnson	F	NM	1970-12-17	49
4	4	Emily Smith	F	TX	1985-03-07	35
5	5	Ashley Smith	F	TX	1975-05-13	45
6	6	Matthew Johnson	M	CA	1984-07-07	35
7	7	Alexis Smith	F	IL	1972-08-16	47
8	8	Megan Wilson	F	CA	1979-04-19	41
9	9	Victoria Davis	F	TX	1983-12-07	36
10	10	Ryan Johnson	M	PA	1976-03-12	44

16.6 习题

【习题16.6】统计每天各优先级别的订单记录数，并将当天高优先级记录数大于低优先级记录数的高优先级统计结果输出到HIGH.txt中。

各优先级文件的部分数据如下：

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
3271	33058	125535.00	1992/1/1	1-URGENT
176768	112850	378570.00	1992/1/1	1-URGENT
201414	11642	170608.00	1992/1/1	1-URGENT
228804	96325	138224.00	1992/1/1	1-URGENT
…	…	…	…	…

1-URGENT.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
23010	91141	97372.00	1992/1/1	2-HIGH
27137	54109	280493.00	1992/1/1	2-HIGH
175975	7511	285691.00	1992/1/1	2-HIGH
249668	115781	39757.00	1992/1/1	2-HIGH
…	…	…	…	…

2-HIGH.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
37543	124229	178835.23	1992/1/1	3-MEDIUM
88199	36158	268020.49	1992/1/1	3-MEDIUM
115585	143335	201582.36	1992/1/1	3-MEDIUM
126753	136505	207936.02	1992/1/1	3-MEDIUM
…	…	…	…	…

3-MEDIUM.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
5607	91021	37356.47	1992/1/1	4-NOT SPECIFIED
20742	136801	121343.42	1992/1/1	4-NOT SPECIFIED
45697	109417	345167.47	1992/1/1	4-NOT SPECIFIED
159266	99056	125324.77	1992/1/1	4-NOT SPECIFIED
…	…	…	…	…

4-NOT SPECIFIED.txt

O_ORDERKEY	O_CUSTKEY	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY
27015	10954	41753.82	1992/1/1	5-LOW
59718	24749	83608.04	1992/1/1	5-LOW
139655	25249	118709.27	1992/1/1	5-LOW
190656	23695	154110.00	1992/1/1	5-LOW
…	…	…	…	…

5-LOW.txt

16.6 习题

解题思路

用joinx函数，实现多个有序游标的连接；用游标函数cs.select筛选结果

	A	B
1	=file("1-URGENT.txt").cursor@t()	/打开文件并创建游标
2	=file("2-HIGH.txt").cursor@t()	/打开文件并创建游标
3	=file("3-MEDIUM.txt").cursor@t()	/打开文件并创建游标
4	=file("4-NOT SPECIFIED.txt").cursor@t()	/打开文件并创建游标
5	=file("5-LOW.txt").cursor@t()	/打开文件并创建游标
6	=A1.groupx(O_ORDERDATE,O_ORDERPRIORITY;count(~):count)	/按订单日期、优先级分组统计订单数
7	=A2.groupx(O_ORDERDATE,O_ORDERPRIORITY;count(~):count)	/按订单日期、优先级分组统计订单数
8	=A3.groupx(O_ORDERDATE,O_ORDERPRIORITY;count(~):count)	/按订单日期、优先级分组统计订单数
9	=A4.groupx(O_ORDERDATE,O_ORDERPRIORITY;count(~):count)	/按订单日期、优先级分组统计订单数
10	=A5.groupx(O_ORDERDATE,O_ORDERPRIORITY;count(~):count)	/按订单日期、优先级分组统计订单数
11	=joinx@1(A6:URGENT,O_ORDERDATE;A7:HIGH,O_ORDERDATE;A8:MEDIUM,O_ORDERDATE;A9:NOT_SPECIFIED,O_ORDERDATE;A10:LOW,O_ORDERDATE)	/对各优先级的聚合结果，按订单日期做对齐式连接
12	=A11.select(HIGH.count>LOW.count)	/筛选高优先级记录数大于低优先级记录数的结果
13	=file("HIGH.txt").export@t(A12.(HIGH))	/将满足要求的结果集中高优先级的结果输出
14	>A11.close()	/关闭游标

润乾
RAQSOFT

THANKS

创新技术 推动应用进步

www.raqsoft.com.cn