

凌云 C++ 编程  
PROGRAMMING

# C++ 信奥编程

(1-2 学阶)

陈琰宏 编著

陈琰宏，副教授，硕士生导师，浙江财经大学人工智能系主任、慧教育研究中心负责人，金牌教练，浙江省学科竞赛优秀指导教师，浙江省中小信息学教材审核专家，原杭州育才集团信奥教练。



在 2021-2023 CSP 信奥竞赛中，所指导的学生获奖 106 人次。近三年，陈老师指导大学生程序设计竞赛获得国际一等奖 2 项、国际二、三等奖二十余项，全国奖三十余项、省级奖百余项；指导蓝桥杯软件设计大赛获得全国一等奖 9 项、全国二等奖。目前，陈老师编写了十余本编程教材与讲义。



# 目 录

## 第一部分 表达式

基本单词 .....	1
框架程序 .....	2
第一讲 认识计算机 .....	3
1.1 计算机的发展简史 .....	3
1.2 计算机系统 .....	4
1.3 信息技术 .....	6
1.4 程序 .....	8
课堂作业列表 .....	9
习题 .....	9
第二讲 初识 C++ .....	11
2.1 C/C++程序的基本框架 .....	11
2.2 编译器 .....	12
2.3 数据输入输出 .....	13
2.4 流程图与算法 .....	14
2.5 上课习题 .....	15
2.6 初学者常见的错误 .....	17
课堂作业列表 .....	18
习题 .....	18
第三讲 变量与表达式 .....	20
3.1 变量 .....	20
3.2 常量 .....	20
3.3 小数点的设置 .....	21
3.4 表达式 .....	22
3.5 类型转换 .....	25
3.6 上课习题 .....	26
课堂作业列表 .....	28
习题 .....	28
第四讲 关系与逻辑 .....	32
4.1 关系运算符及关系表达式 .....	32
4.2 逻辑运算符 .....	32
4.3 运算符的优先级和结合性 .....	34
4.4 *常用的数学函数 .....	34
4.5 上课习题 .....	35

## 第二部分 选择

第五讲 IF 语句 .....	41
5.1 C/C++的语句 .....	41
5.2 单分支结构 .....	42
5.3 双分支结构 .....	43
5.4 *条件运算符和条件表达式 .....	44
5.5 案例讲解 .....	45
课堂作业列表 .....	47
习题 .....	47
第六讲 多路选择 .....	50
6.1 IF 多路选择 .....	50
6.2 案例分析 .....	51
课堂作业列表 .....	53
习题 .....	53
第七讲 switch 语句 .....	58
7.1 switch 语句 .....	58
7.2 案例分析 .....	58
习题 .....	62
第八讲 推理 .....	65
1 [7599] 破案 .....	65
2 [7853] 围棋赛 2 .....	66
3[7628] 数学竞赛 .....	67
4 [7629] 苹果数量* .....	68
5 [7741]地理课* .....	68
习题 .....	69
第九讲 循环初探 .....	73
1.1 知识剖析 .....	73
1.2 案例分析 .....	74
第十讲 表达式综合 .....	79
1 [7743] 阶梯电价 .....	79
2 [2164] 计算个人所得税 .....	80
3[3228] 小 Q 买笔 .....	81
4 [2613] 点和正方形的关系 .....	82

## 第三部分 循环

1 while 循环 .....	83
1.1 知识剖析 .....	83
1.2 案例分析 .....	84
1[7124] 计算好天气 .....	84
2[2195]求和 .....	85
3 [2206] 求各位数字之和 .....	87

4 [1048] 统计输出人数 .....	88
6 [7708] 停车收费 .....	89
1.3 思维拓展：等差序列 .....	91
7 [1072] 等差数列 .....	91
2 等差序列 .....	94
1.1 知识剖析 .....	94
2.2 案例分析 .....	95
1[2540] 等差数列末项计算 .....	95
2 [7619] 等差数列 2 .....	95
3 [7620] 计算求和 .....	96
4 [7621] 数座位数 .....	96
5 [7622] 青蛙跳 .....	97
2.3 思维拓展：奥运奖牌 .....	98
6 [2622] 奥运奖牌计数 .....	98
3 do..while 循环 .....	102
3.1 知识剖析 .....	102
1[2195] 求和 .....	102
3.2 案例分析 .....	103
2 [1050] 偶数和 .....	103
3[2647] 级数求和 .....	104
4 [7616] 求 01 数字和 .....	105
5[2646] 角谷猜想 .....	106
6 [1335] 水仙花数 .....	107
3.3 思维拓展：统计 .....	108
7 [7617] 分数统计 .....	109
4 for 循环 .....	114
4.1 知识剖析 .....	114
1[2195] 累加和 .....	115
2[1019] 累加求和 .....	116
3 [2617] 均值 .....	116
4.2 案例分析 .....	117
4 [2198] 求最小值 .....	117
5 [2620] 最大跨度值 .....	117
6[1074] 计负均正 .....	118
4.3 思维拓展：环形路行程问题 .....	119
7 [7618] 环形路行程问题 .....	119
5 平均数 .....	123
5.1 知识剖析 .....	123
1 [2862] 求平均分 .....	123
2 [1514] 平均分 .....	124
3 [7623] 5 班平均分 .....	125
4 [1312] 十佳歌手大赛 .....	125
5.2 案例分析 .....	126
5[7624] 春蕾小队爬山 .....	126

6 [7625] 服装厂 .....	127
7[7626] 改动数据 .....	128
5.3 思维拓展：数字翻转 .....	128
习题 .....	130
6 循环嵌套 .....	133
6.1 [1436]阶乘和 .....	133
6.2 完数 .....	134
2 [7140] 判断完数 .....	134
3 [2251] 完数 .....	135
6.3 案例分析 .....	136
4 [2655] 数 1 的个数 .....	136
5 [2656] 数字统计 .....	136
6.4 思维拓展：Fibonacci 数 .....	137
6 [2631] 菲波那契数 .....	137
7 [2321]: 斐波那契数列求和 .....	139
8[7338] 数字蜂房 .....	139
7 序列求和 .....	144
7.1 归纳 .....	144
7.2 案例分析 .....	144
1[2623] 奇数求和 .....	144
2 [2653] 计算多项式的值 .....	145
3 [1021] 前 N 项和 .....	146
4 [2122]多项式求和 2 .....	146
5 [2652] 求出 e 的值 .....	147
6 [1437] 三数的和 .....	147
7.3 思维拓展：金币 .....	148
7[2660] 金币 .....	148
8 鸡兔同笼 .....	154
8.1 知识剖析 .....	154
1 [2200] 鸡兔同笼 .....	154
8.2 案例解析 .....	155
2 [3309] 正整数解 .....	155
3 [7632] 三轮摩托车 .....	156
4[7633]大小猴子 .....	157
5 [7740] 蜘蛛蜻蜓 .....	158
6 [2655]数 1 的个数 .....	159
8.3 思维拓展：百钱买百鸡 .....	160
7 [3549]：桐桐去购物 .....	160
9 break/continue .....	164
9.1 break 语句 .....	164
1[6854]与 7 无关的数 .....	165
2 [1048] 统计输出人数 .....	166
3 [2121] 多项式求和 .....	167
4 [1481]数列求和 .....	168

9.2 continue 语句 .....	169
5 [2654] 与 7 无关的数 .....	170
6 [3324] 小鱼的航程 .....	171
9.3 思维拓展：素数 .....	171
2 [3199] 素数判断 .....	171
10 质数 .....	176
10.1 知识剖析 .....	176
1 [3199] 素数判断 .....	178
10.2 案例解析 .....	178
2 [2253]素数的个数与和 .....	178
3 [3304] 素数-小数据 .....	179
4 [7631] 素数-大数据 .....	180
5[2658] 质因数分解 .....	180
6 [3556] 孪生素数 .....	181
9.4 思维拓展 函数 .....	182
课堂作业列表 .....	183
11 逻辑推理 .....	188
11.1 知识剖析 .....	188
1 [7143] 破案 .....	188
2 [2221] 推算车牌号码 .....	189
11.2 案例解析 .....	190
3 [7627] 围棋赛 .....	190
4[7628] 数学竞赛 .....	191
5 [7629] 苹果数量 .....	192
6 [7741]地理课 .....	193
7 [3177] 谁参赛了 .....	194
11.3 思维拓展：数组 .....	196
12 图形绘制 1 .....	200
1[3229] 输入图形 1 .....	200
2[7397] 简单图形 .....	200
3[1533] 树形图案 .....	202
4[7180] 沙漏图形 .....	203
5 [2657] 画矩形 .....	204
6 [2365] 奥巴马会编程 .....	206
7[1330] 打印空心等腰梯形 .....	207
13 图形绘制 2 .....	212
1 [1023]叠字母 .....	212
2 [7181] 层奇数三角形 .....	213
3 [1024]图形输出 .....	214
4[1503] ASCII 艺术 .....	216
思维拓展 .....	217
5[2126] 打印汉字 .....	217
课堂作业列表 .....	218
课前练习 .....	218

14 数学拓展 .....	223
14.1 公约数和最小公倍数 .....	223
1[1433] 公约数和最小公倍数 .....	223
2[3044] 公约数和最小公倍数 2 .....	224
14.2 环形路行程问题 .....	224
3 [7618] 环形路行程问题 .....	225
14.3 分解质因数 .....	226
4 [1329]: 质数因子分解 .....	226
4 [3552] 桐桐的数学难题 .....	227
13.4 完数 .....	228
课堂作业列表 .....	228
课前练习 .....	228
课后练习 .....	228
15 综合练习 .....	229
[1] 1328: 校园健身走 .....	229
[2]1553: 生活费 X1 .....	230
3 [7314] 排座位 .....	231
4[7316] 步步夺金 .....	233
5[7351] 报数游戏 .....	235
6[7332] 最佳时间 .....	236
7[7354] 换座位 .....	237



## 基本单词

int 整数

char 字符

cin/scanf 输入

main 主函数

include 包含

namespace 命名空间

cmath 数学

abs 取绝对值

double 小数（实数）

bool 布尔（1 或 0）

cout /printf 输出

return 范围

using 使用

iostream 输入输出流

sqrt 开根号

strlen 取字符串长度

## 框架程序

```
#include <iostream>                //预处理命令
using namespace std;                //使用命名空间 std
int main( )                          //主函数首部
{                                    //函数体开始
    int a,b,sum;                    //定义变量
    cin>>a>>b;                      //输入语句
    sum=a+b;                        //赋值语句
    cout<<sum<<endl;                //输出语句
    return 0;                       //程序结束，返回一个零值
}                                    //函数结束

scanf("%d %d",&a,&b);                //输入语句

printf("%d",sum);                   //输出语句
```

# 第一讲 认识计算机

主要内容：

1. 了解计算机发展史、计算机系统的组成。
2. 理解编程环境。
3. 理解二进制

## 1.1 计算机的发展简史

### 一、计算机的诞生

1945年，美籍匈牙利科学家冯·诺依曼提出一个电子计算机设计方案，其中提到了两个设想：采用二进制和“存储程序”，并能够自动地执行程序。这两个设想对于现代计算机至关重要，也使冯·诺依曼成为“现代电子计算机之父”，冯·诺依曼机体系延续至今。

依据冯·诺依曼的设计思路，1946年2月15日，“莫尔小组”研制成功世界上第一台通用数字电子计算机ENIAC，承担开发任务的莫尔小组”由四位科学家和工程师埃克特、莫克利、戈尔斯坦、博克斯组成，总工程师埃克特当时年仅24岁。

ENIAC长30.48米，宽6米，高2.4米，占地面积约170平方米，30个操作台，重达30英吨，耗电量150千瓦，造价48万美元。它包含了17,468根真空管（电子管）7,200根晶体二极管，1,500个中转，70,000个电阻器，10,000个电容器，1500个继电器，6000多个开关，计算速度是每秒5000次加法或400次乘法，是使用继电器运转的机电式计算机的1000倍、手工计算的20万倍。

虽然ENIAC体积庞大，耗电惊人，运算速度不过几千次，但它比当时已有的计算装置要快1000倍，而且还有按事先编好的程序自动执行算术运算、逻辑运算和存储数据的功能。ENIAC宣告了一个新时代的开始。从此科学计算的大门也被打开了。

艾伦·麦席森·图灵是另一位伟大的先驱，被称为计算机科学之父，人工智能之父。图灵提出了一种用于判定机器是否具有智能的试验方法，即图灵试验，至今，每年都有试验的比赛。此外，图灵提出的著名的图灵机模型为现代计算机的逻辑工作方式奠定了基础。



冯·诺依曼

### 二、计算机在我国的发展

我国在1957年开始研制通用数字电子计算机，并于1958年8月1日，诞生了我国第一台电子计算机，该机可以表演短程序运行。在研制第一代电子管计算机的同时，我国已开始研制晶体管计算机，1965年研制成功的我国第一台大型晶体管计算机（109乙机）。两年后又推出109丙机，它运行了15年，有效算题时间10万小时以上，在我国两弹试验中发挥了重要作用，被誉为“功勋机”。

IBM公司在1964年推出了360系列大型机，这是美国进入第三代计算机时代的标志，而我国到1970年初期才陆续推出采用集成电路的计算机。在1973年，北京大学与北京有线电厂等单位一起合作，研制运算速度每秒100万次的大型通用计算机。

与国外一样，我国第四代计算机研制也是从微机开始的。1983年，电子部六所研制成功与IBM PC机兼容的DJS-0520微机。1992年，国防科大研究成功通用并行巨型机：银河-II，1997年国防科大研制成功银河-III百亿次并行巨型计算机系统，采用可扩展分布共享存储并行处理体系结构，由130多个处理结点组成，峰值性能为每秒130亿次浮点运算，系统综合技术达到90年代中期国际先进水平。2001年我国研制的“曙光”巨型计算机的速度更是超过了每秒4000亿次。我国成为继美国、日本之后世界上第三个具备研制高性能计算机能力的国家。



2018年“神威·太湖之光”超级计算机系统其实测运算速度可以达到每秒9.3亿亿次，各项指标雄居2018年世界第一。2019年11月19日，新一期全球超级计算机500强榜单面世，美国超级计算机“顶点”蝉联冠军，中国则继续扩大数量上的领先优势，在总算力上与美国的差距进一步缩小。美国超级计算机“顶点”以每秒14.86亿亿次的浮点运算速度再次登顶，第二位是美国超算“山脊”，中国超算“神威·太湖之光”和“天河二号”分列三、四位。

## 1.2 计算机系统

计算机系统包括计算机硬件系统和软件系统。硬件系统遵循冯·诺依曼计算机结构模型的基本思想，由运算器、控制器、存储器、输入设备和输出设备五大基本部件组成。软件系统一般指为计算机运行工作服务的全部技术和各种程序。软件系统由系统软件和应用软件两大部分组成。

### 1.2.1 硬件系统

计算机硬件是计算机的重要组成部分，其中包含了5个重要的组成部分：运算器、控制器、存储器、输入设备、输出设备。

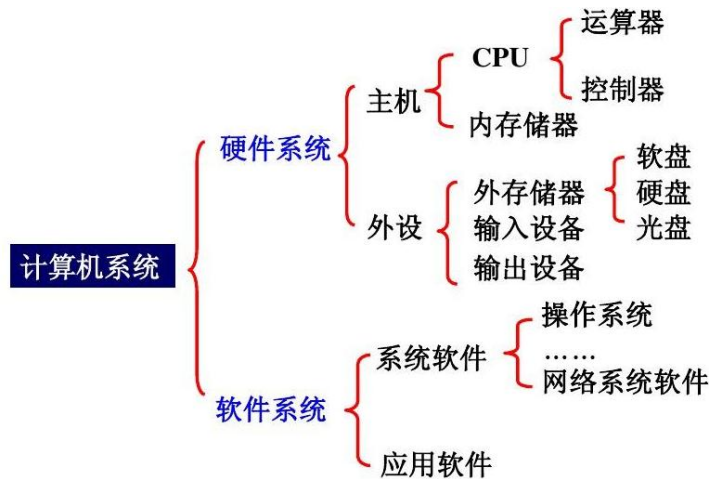
**1 运算器** 计算机硬件中的运算器主要功能是对数据和信息进行运算和加工。运算器包括以下几个部分：通用寄存器、状态寄存器、累加器和关键的算术逻辑单元。运算器可以进行算术计算（加减乘除）和逻辑运算（与或非）。

**2 控制器** 控制器和运算器共同组成了中央处理器（CPU）。控制器可以看作计算机的大脑和指挥中心，它通过整合分析相关的数据和信息，可以让计算机的各个组成部分有序地完成指令。

**3 存储器** 顾名思义，存储器就是计算机的记忆系统，是计算机系统中的记事本。而和记事本不同的是，存储器不仅可以保存信息，还能接受计算机系统内不同的信息并对保存的信息进行读取。存储器由主存和辅存组成，主存就是通常所说的内存，分为RAM和ROM两个部分。辅存即外存，但是计算机在处理外存的信息时，必须首先经过内外存之间的信息交换才能够进行。

**4 输入设备** 输入设备和输出设备都是进行人机互动的关键设备。鼠标、键盘等输入设备的出现，给计算机带来了天翻地覆的变化。现有的鼠标主要有两类：光电鼠标和机械式鼠标。通过鼠标，我们可以很方便地在计算机屏幕上进行坐标的定位，可以很好地操作图形和软件处理，为人类提供了最大的便捷。键盘也是一类非常重要的输入设备，计算机大部分的指令都是通过键盘输入来进行的。

**5 输出设备** 输出设备也是计算机人机互动的关键设备，它的特点是可以将计算机的信息以画面的形式展现出来，具有很好的直观性。常见的输出设备有显示器、打印机、语音和视频输出装置等。



## 1.2.2 计算机软件系统

计算机的软件系统是指计算机在运行的各种程序、数据及相关的文档资料。**计算机软件系统通常被分为系统软件和应用软件两大类。**计算机系统软件能保证计算机按照用户的意愿正常运行，为满足用户使用计算机的各种需求，帮助用户管理计算机和维护资源执行用户命令、控制系统调度等任务。软件系统分为两大类，虽然各自的用途不同，但他们的共同点是都存储在计算机存储器中，以某种格式编码书写的程序或数据。

### (1) 系统软件

系统软件是指担负控制和协调计算机及其外部设备、支持应用软件的开发和运行的一类计算机软件。系统软件一般包括操作系统、语言处理程序、数据库系统和网络管理系统。

### (2) 应用软件

应用软件是指为特定领域开发、并为特定目的服务的一类软件。应用软件是直接面向用户需要的，它们可以直接帮助用户提高工作质量和效率，甚至可以帮助用户解决某些难题。应用软件一般分为两类：一类是为特定需要开发的实用型软件，如会计核算软件、工程预算软件和教育辅助软件等；另一类是为了方便用户使用计算机而提供的一种工具软件，如用于文字处理的 Word、用于辅助设计的 AutoCAD 及用于系统维护的瑞星杀毒软件等。



## 1.3 信息技术

现代信息技术主要的技术是感测、通信、计算机、控制等技术，在这些技术的实现过程中，又离不开微电子技术的应用。

信息技术的主要领域为：

- 微电子与光电子技术
- 新型电子元器件和电子信息材料技术
- 现代通信技术
- 计算机网络技术
- 智能检测与控制技术

### 1.3.1 IT 领域的三大定律

第一定律：摩尔定律

美国人高登·摩尔提出摩尔定律。集成电路上可容纳的晶体管数，每隔一年半左右就会增加一倍，性能也提高一倍。即微处理器的速度每 18 个月翻一翻。

第二定律：吉尔德定律

在未来 25 年，主干网的带宽每 6 个月增长一倍，其增长速度是摩尔定律预测的 CPU 增长速度的 3 倍。吉尔德定律的提出者是被称为“数字时代三大思想家”之一的乔治·吉尔德。

第三定律：麦特卡尔夫定律

网络的价值同网络用户数量的平方成正比，也就是说，N 个联结创造出 N\*N 的效益。该定律的提出者为以太网的发明人鲍勃·麦特卡尔夫。

### 1.3.2 二进制及其运算

常用的是十进位计数制，简称十进制；就是按照“逢十进一”的原则进行计数的。在实际应用中，还使用其它的计数制，如二进制（两只鞋为一双）、十二进制（十二个信封为一打）、二十四进制（一天 24 小时）、六十进制（60 秒钟为一分钟，60 分钟为一小时）等等十进制大家都很熟悉，例如，一台 mp4 的价格是 399.58 元，我们会自动地反映成：

$$3 \times 10^2 + 9 \times 10^1 + 9 \times 10^0 + 5 \times 10^{-1} + 8 \times 10^{-2} \text{ 元}$$

#### 一、其他进制转化为十进制

由于任一数都可以按权展开，于是很容易将一个非十进制数转换为相应的十进制数。具体的步骤是：将一个非十进制数按权展开成一个多项式，每项是该位的数码与相应的权之积，把多项式按十进制数的规则进行求和，所得结果即是该数的十进制

$$1101.11B = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 13.75$$

$$111010.1B = 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1}$$

$$= 32 + 16 + 8 + 2 + 0.5 = 58.5$$

$$324.6Q = 3 \times 8^2 + 2 \times 8^1 + 4 \times 8^0 + 6 \times 8^{-1} = 212.75$$

$$D4.CH = 13 \times 16^1 + 4 \times 16^0 + 12 \times 16^{-1} = 212.75$$

## 二、十进制转化为二进制

例：将  $(35.6875)_{10}$  转换为二进制数。

① 用除 2 取余法将整数部分  $(35)_{10}$  转换为二进制整数：

2	35	..... 余数为1	↑ 低位
2	17	..... 余数为1	
2	8	..... 余数为0	
2	4	..... 余数为0	
2	2	..... 余数为0	
2	1	..... 余数为1	↓ 高位
	0		

故： $(35)_{10} = (100011)_2$

验证： $1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 32 + 2 + 1 = 35$

用乘 2 取整法将小数部分  $(0.6875)_{10}$  转换为二进制形式：

<b>0.6875</b>		
× 2		
<b>1.3750</b>	..... 整数部分为1	↑ 高位
<b>0.3750</b>		
× 2		
<b>0.7500</b>	..... 整数部分为0	
<b>0.7500</b>		
× 2		
<b>1.5000</b>	..... 整数部分为1	
<b>0.5000</b>		
× 2		
<b>1.0000</b>	..... 整数部分为1	↓ 低位

即： $(0.6875)_{10} = (0.1011)_2$

## 三、八进制和二进制之间的转换

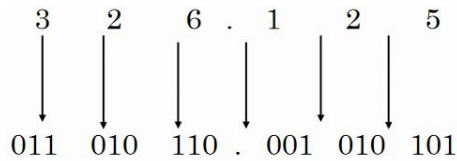
由于二进制数与  $2^k$  进制数的基数都是 2 的幂次，所以二进制数的 k 位对应于  $2^k$  进制数的一位。对于非整数的二进制数应分成整数部分与纯小数部分分别加以转换。

八进制的基数  $R=8=2^3$ ，必须用三位二进制数来构成一位八进制数码，采用分组对应转换法。

转换方法：将二进制数转换成八进制数时，首先从小数点位置开始，向左或向右将二进制数的整数和小数部分每三位分为一组，不足三位的分别在整数的最高位前和小数的最低位后加“0”补足，然后写出每一组二进制数所对应的八进制数码即可。

例：  $(10\ 100.101\ 1)_2 \rightarrow (?)_8$

例：  $(326.125)_8 \rightarrow (?)_2$



### 1.3.3 数据存储

字节 (Byte) 是存储数据的基本单位，并且是硬件所能访问的最小单位。内存中存储数据的最小单位是“位 (Bit)”。字节是存储数据的基本单位，位是存储数据的最小单位。

内存里面存放的全是二进制代码。内存里面有很多“小格子”，每个“格子”中只能存放一个 0 或 1。一个“小格子”就是一位，所以“位”要么是 0。8 个“小格子”就是一字节，即一字节等于 8 位。

常见的存储单位主要有 bit (位)、B (字节)、KB (千字节)、MB (兆字节)、GB (千兆字节)。它们之间主要有如下换算关系：

1B=8bit                      1KB=1024B                      1MB=1024KB                      1GB=1024MB

其中 B 是 Byte 的缩写。

比如计算机的内存是 4GB，那么它能存放多少个 0 或 1 (即能存放多少位) 呢？  
 $4 \times 1024 \times 1024 \times 1024 \times 8$  位。

## 1.4 程序

所谓程序就是用程序设计语言描述的、用于控制计算机完成某一特定任务的程序设计语言语句的集合。语句是程序设计语言中具有独立逻辑含义的单元，它可以分解为一条计算机指令，也可以分解为若干条计算机指令的集合。人们通过编写程序，发挥计算机的优势，帮助解决各种问题。

编写计算机程序所用的语言是人与计算机之间交流的工具，一般可分为机器语言、汇编语言和高级语言。

#### 1、机器语言 (Machine Language)

机器语言是计算机系统所能识别的，不需要翻译直接供机器使用执行的程序设计语言。

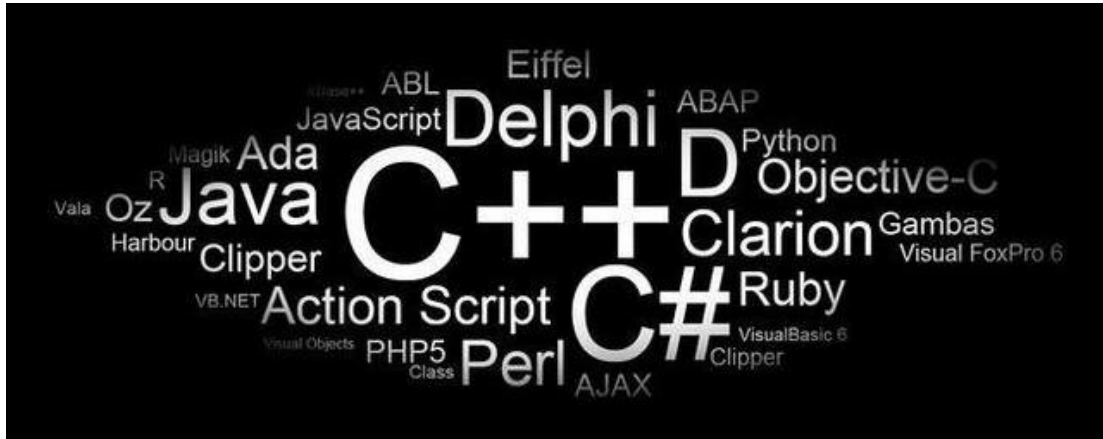
#### 2、汇编语言 (Assemble Language)

汇编语言是一种面向机器的程序设计语言，它是为特定的计算机设计的。

#### 3、高级语言

比较接近于人们习惯用的自然语言和数学表达式，因此称为高级语言。高级语言包括很多编程语言，如目前流行的 c, c++, C#, pascal, java, python, lisp, FoxPro, VC, 等。这些语言的语法、命令格式都不相同。





标准 C++ 程序格式如下：

例 1.1 从键盘上输入两个数据 a 和 b，计算 a 和 b 之和并输出

```

#include <iostream>           //预处理命令
using namespace std;        //使用命名空间 std
int main( )                 //主函数首部
{                             //函数体开始
    int a,b,sum;            //定义变量
    cin>>a>>b;              //输入语句
    sum=a+b;                //赋值语句
    cout<<sum<<endl;        //输出语句
    return 0;                //如程序正常结束，向操作系统返回一个零值
}                             //函数结束

```

## 课堂作业列表

1000 求两数之和 计算 a+b	2557 Hello,World!
2558 输出第二个整数	1002 交换两个变量 例题 1.4
2560 字符三角形	3866 静夜思
6873 输出年龄	

## 习题

- 1.一般认为，世界上第一台电子数字计算机诞生于( )  
A. 1946 年 B. 1952 年 C. 1959 年 D. 1962 年
- 2.当前的计算机一般被认为是第四代计算机，它所采用的逻辑元件是( )  
A. 集成电路 B. 晶体管 C. 大规模集成电路 D. 电子管
- 3.下列度量单位中，用来度量 CPU 时钟主频的是( )



## 第二讲 初识 C++

主要内容：

1. 理解程序框架
2. 掌握输入 cout
3. 理解编译器
4. 理解流程图

C++是 C 语言的继承，它既可以进行 C 语言的过程化程序设计，又可以进行面向对象的程序设计。C++不仅拥有计算机高效运行的实用性特征，同时还致力于提高大规模程序的编程质量与程序设计语言的问题描述能力。

世界上第一种计算机高级语言是诞生于 1954 年的 FORTRAN 语言。之后出现了多种计算机高级语言。1970 年，AT&T 的 Bell 实验室的 D.Ritchie 和 K.Thompson 共同发明了 C 语言。研制 C 语言的初衷是用它编写 UNIX 系统程序，因此，它实际上是 UNIX 的“副产品”。它充分结合了汇编语言和高级语言的优点，高效而灵活，又容易移植。

1971 年，瑞士联邦技术学院 N.Wirth 教授发明了 Pascal 语言。Pascal 语言语法严谨，层次分明，程序易写，具有很强的可读性，是第一个结构化的编程语言。

1979 年，Bjame Sgoustrup 到了 Bell 实验室，开始从事将 C 改良为带类的 C(C with classes) 的工作。1983 年该语言被正式命名为 C++。自从 C++被发明以来，它经历了 3 次主要的修订，每一次修订都为 C++增加了新的特征并作了一些修改。第一次修订是在 1985 年，第二次修订是在 1990 年，而第三次修订发生在 c++的标准化过程中。在 20 世纪 90 年代早期，人们开始为 C++建立一个标准，并成立了一个 ANSI 和 ISO (International Standards Organization) 国际标准化组织的联合标准化委员会 1998 年，C++的 ANSI/ISO 标准被投入使用。通常，这个版本的 C++被认为是标准 C++。所有的主流 C++编译器都支持这个版本的 C++。

C++标准第三版，2011 年 8 月 12 日发布。C++标准第四版，2014 年 8 月 18 日发布。

### 2.1 C/C++程序的基本框架

如下代码所示，为 C++程序的框架程序

<p>框架程序 1：直接输出</p> <pre>#include &lt;iostream&gt; //预处理命令 using namespace std; //使用命名空间 std int main() //主函数首部 { //函数体开始     cout&lt;&lt;"Hello World"&lt;&lt;endl; //输出语句     return 0; //程序结束返回一个零值 } //函数结束</pre>	<p>框架程序 2：定义变量求解输出</p> <pre>#include &lt;iostream&gt; using namespace std; int main() { //函数体开始     int a,b,sum; //定义变量     cin&gt;&gt;a&gt;&gt;b; //输入语句     sum=a+b; //赋值语句     cout&lt;&lt;sum&lt;&lt;endl; //输出语句     return 0; }</pre>
--	---

框架程序分头文件和主函数，主函数包括 5 个部分：定义、输出或初始化、求解、输出、返回值。

头文件：

```
#include <iostream>           //预处理命令
using namespace std;         //使用命名空间 std
```

定义：

```
int a,b,sum;                 //定义变量
```

输入：

```
cin>>a>>b;                  //输入语句
```

求解：

```
sum=a+b;                    //赋值语句
```

输出：

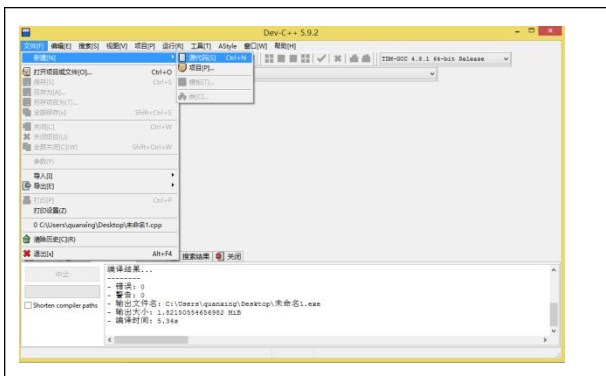
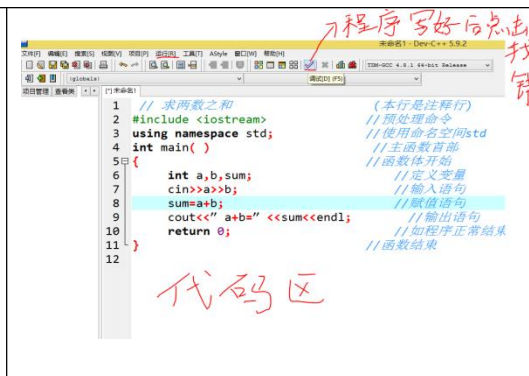
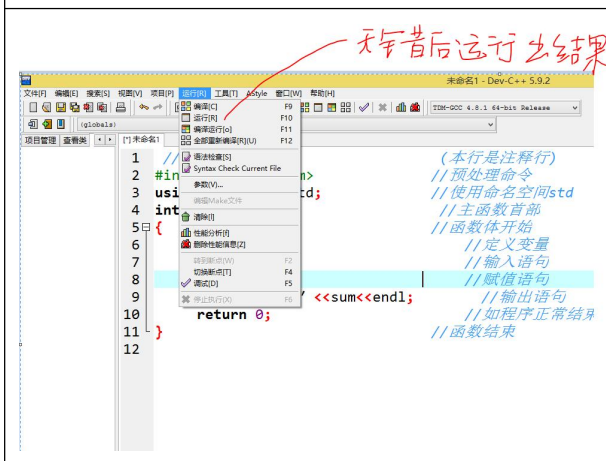
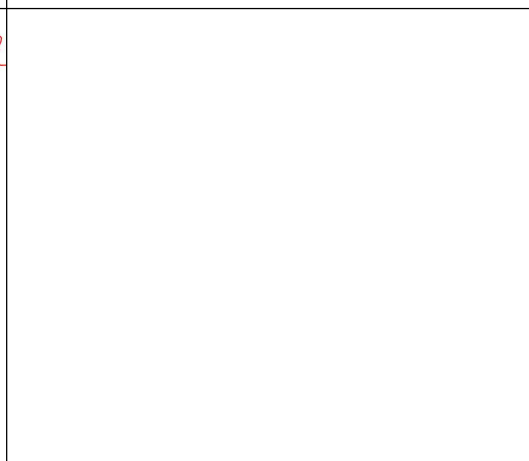
```
cout<<sum<<endl;           //输出语句
```

返回值：

```
return 0;
```

## 2.2 编译器

程序就如同用英语（程序设计语言）写作的文章，要让一个懂得英语的人（编译器）来阅读、理解、标记这篇文章。一般的，以英语文本为基础的计算机程序要经过编译、链接而成为人难以解读，但可轻易被计算机所解读的数字格式，然后放入运行。这个过程被称为编译。

## 2.3 数据输入输出

C++语言输入输出是调用标准库函数中的输入输出函数来实现的。在使用时，应在源程序的开头使用如下语句：

```
#include<iostream>
using namespace std;
```

C++语言标准函数库提供了许多标准输入、输出函数，常用的6个最基本的输入、输出函数：流输入 cin，流输出 cout，字符输入 `getchar`，字符输出 `putchar`，格式化输入 `scanf`，格式化输出 `printf`。

### 1.3.1 cout 流输出数据

`cout` 语句是 C++ 中实现输出功能的语句。其格式为：

```
cout << 项目 1 << 项目 2 << ... << 项目 n;
```

如果项目是表达式，则输出表达式的值；如果项目加引号，则输出引号内的内容；如果项目是 `endl`，则表示换行。

#### 1) 输出字符串

每当我们输出字符串常量的时候，必须用双引号把字符串引起来，以便将它和变量名明显的区分开来。

下面两个语句是不同的：

```
cout << "Hello";           //打印字符串 Hello
cout << Hello;             //把变量 Hello 存储的内容打印到屏幕上
```

#### 2) 换行符的使用

必须注意，除非我们明确指定，`cout` 并不会自动在其输出内容的末尾加换行符，因此下面的语句：

```
cout << "This is a sentence.";
cout << "This is another sentence.";
```

将会有如下内容输出到屏幕：

```
This is a sentence.This is another sentence.
```

可以用操作符 `endl` 来换行，例如：

```
cout << "First sentence." << endl;
cout << "Second sentence." << endl;
```

将会输出：

```
First sentence.
Second sentence.
```

#### 3) 增强信息的可读性

为了增强输出信息的可读性，在输出多个数据时可以通过插入空格符，换行符或其他提示信息将数据进行组织，以获得更好的效果

```
x=12;
cout << "tom is my friend,he is";
cout << x;
```

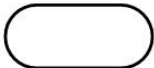
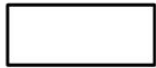

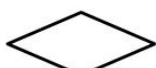


例 1 阅读并上机调试以下程序，体会 cout 语句的使用。

```
#include<iostream>
using namespace std;
int main(){
    cout << 2 + 4 << endl;
    cout << "ABCD" << endl;
    int a = 10,b = 5;
    cout << a + b << endl;
    return 0;
}
```

## 2.4 流程图与算法

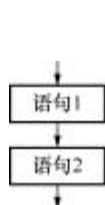
人们使用计算机，就是要利用计算机处理各种不同的问题，而要做到这一点，人们就必须事先对各类问题进行分析，确定解决问题的具体方法和步骤，再编制好一组让计算机执行的指令即程序，

算法是解决一个问题采取的方法和步骤的描述，流程图是传统的算法表示方法，它使用不同的几何图形框来代表各种不同性质的操作，用流程线来指示算法的执行方向。由于它直观形象，易于理解，所以应用广泛。下图所示为流程图中的主要符号。

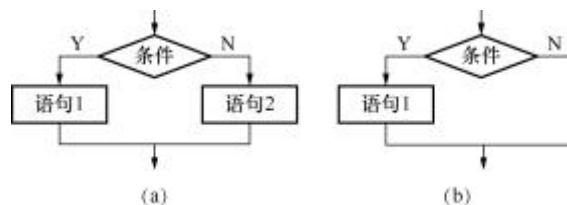
符号	含义	符号	含义
	起止框，表示算法的开始和结束		处理框，表示初始化或赋值等操作
	输入输出框，表示数据的输入输出操作		判断框，表示根据一个条件决定执行两种不同操作中的其中一个
	流程线，表示流程的方向		连接点，用于流程的分页连接

算法通常用循序、选择与循环三种基本结构表示：

1) 顺序结构。顺序结构是简单的线性结构，各框按顺序执行。其流程图如图所示，语句的执行顺序为：语句 1→语句 2。一般而言，顺序结构的算法中应包括几个基本操作步骤：确定求解过程中使用的变量、变量类型和变量的值；按算法进行运算处理；输出处理结果。



顺序结构



选择结构

2) 选择结构。这种结构是对某个给定条件进行判断，条件为真或假时分别执行不同的框的内容。其基本形状有两种，如图所示，执行序列为：当条件为真时执行语句 1，否则执

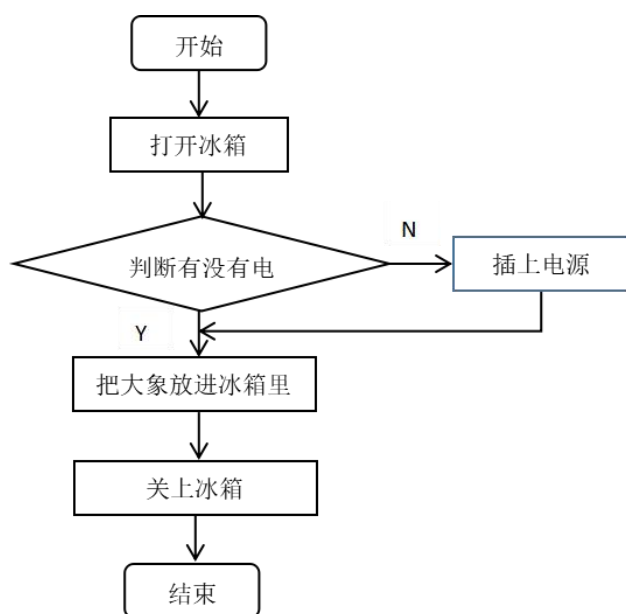
行语句 2。图 (b) 的执行序列为：当条件为真时执行语句 1，否则什么也不做。

3) 循环结构。循环结构实现某些语句的重复执行。执行时，通常先判断条件，当条件为真时，反复执行“语句组”（也称循环体），一旦条件为假，跳出循环，执行循环的后继语句。

例如顺序结构，顺序结构程序是按照语句的先后顺序依次执行语句的程序。

### 例 2：把大象放冰箱里的算法，分几步？

1. 打开冰箱
2. 看看有没有电
3. 有电就把大象放进冰箱里，没电就电源
4. 关上冰箱的门



## 2.5 上课习题

### 1[ 3294] 春晓

题目描述

春晓

作者：孟浩然

春眠不觉晓，处处闻啼鸟。

夜来风雨声，花落知多少。

现在请你输出这首诗。

```
#include <iostream>
using namespace std;
int main()
```

```

{
    cout<<"春眠不觉晓，处处闻啼鸟。"<<endl;
    cout<<"夜来风雨声，花落知多少。";
    return 0;
}

```

## 2 [6873] 输出年龄

输入一个变量  $n$ , 表示年龄  
 输出一串包含年龄的字符串

参考程序:

样例输入

7

样例输出

I am 7 years old.

```

#include<iostream>
using namespace std;
int main()
{

    int n;
    cin>>n;
    cout<<"I am "<<n<<" years old.";
    return 0;
}

```

## 3 [流程图] 计算好天气

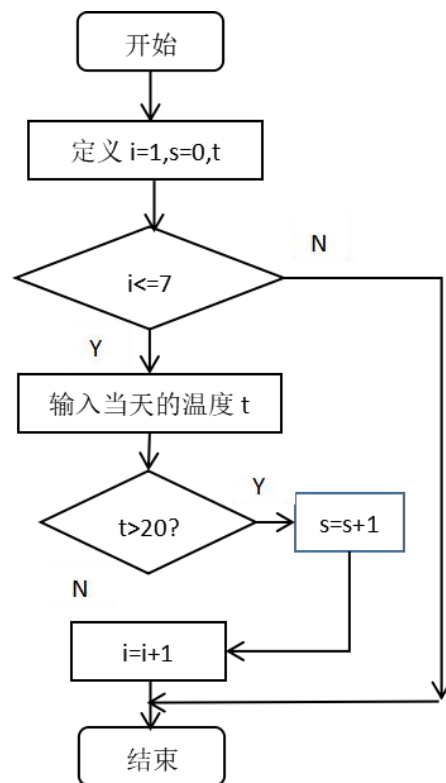
如果认为温度大于 20 度就是好天气，输入周一到周日的每天的温度，统计好天气的天数。输入 7 天的温度输出好天气的天数

样例输入 21 18 19 23 22 16 23

样例输出 4

步骤:

1. 定义 温度、天数、好天气的天数等
2. 循环判断 7 的温度
  - (1) 输入当天温度
  - (2) 判断该温度是否大于 20
  - (3) 如果大于 20 则进行计数





## 2.6 初学者常见的错误

1. 每句话结束的 ; 没有写。注意: () <> 后面没有;

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int a,b;
6     cin>>a>>b;
7     cout<<a+b<<endl;
8     return 0;
9 }
```

C:\Users\win10\Desktop\未命名1.cpp

[Error] expected ';' before 'cout'

[Error] expected ';' before 'cout'

2. 拼写错误

```
1 #include<iostream>
2 using namespace std;
3 int mian()
4 {
5     int a,b;
6     cin>>a>>b;
7     cout<<a+b<<endl;
8     return 0;
9 }
```

C:\Users\win10\Desktop\未命名1.cpp

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int a,b;
6     cin>>a>>b;
7     cont<<a+b<<endl;
8     return 0;
9 }
```

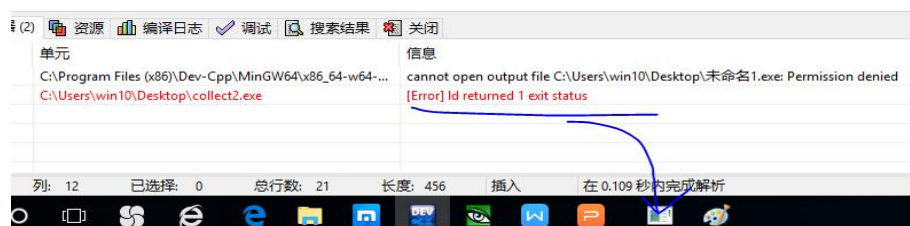
[Error] 'cont' was not declared in this scope

3. 英文符号写成中文符号

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int a,b;
6     cin>>a>>b;
7     cont<<a+b<<endl;
8     return 0;
9 }
```

4. [Error] ld returned 1 exit status

还有 exe 文件在运行



## 课堂作业列表

2564 计算(a+b)*c 的值	2565 计算(a+b)/c 的值
2566 带余除法	2593 A*B 问题
2567 计算分数的浮点数值	3230 成绩
2323 黑鸡跑 1000	4019 求梯形面积
2861 输出对调后的数	

## 习题

1. 请编写程序，输出王维的《竹里馆》诗。

独坐幽篁里，弹琴复长啸。

深林人不知，明月来相照。

```
_____ 1
_____ 2 //头文件
int main(){
    cout<<"独坐幽篁里， ";
    _____ 3
    cout<<"深林人不知， ";
    cout<<"明月来相照。"<<endl;
    return 0;
}
```

2. 从键盘上输入两个数据 a 和 b，计算 a 和 b 之和并输出

```
#include <iostream>
using namespace std;
int main()
{
    _____ //定义 a,b
    cin >> a >> b;//输入 a,b
    cout << a+b << endl;
    return 0;
}
```

3. 天安门，坐落在中华人民共和国首都北京市的中心、故宫的南端，与天安门广场以及人民英雄纪念碑、毛主席纪念堂、人民大会堂、中国国家博物馆隔长安街相望。请输入天安门广场的长和宽，试编一程序，算算天安门广场面积是多少平方米？

```
#include <iostream>
using namespace std;
int main(){
    _____ 1 //定义长和宽
    _____ 2 //输入长和宽
    _____ 3 //输出面积
```

```
    return 0;
}
```

4. 给定 3 个整数 a、b、c，计算表达式(a+b)\*c 的值。

```
#include<iostream>
1
int main(){
2
cin>>a>>b>>c;
3
}
```

5. 一个农夫带着一只狼、一只羊和一棵白菜，要过河。过河条件是，狼不能和羊单独在一起，羊不能和白菜单独在一起。每次过河农夫的船只能带一样东西。请设计过河方案，画出流程图。

# 第三讲 变量与表达式

主要内容:

1. 理解变量与常量
2. 掌握 C/C++的基本数据类型:int ,double, char
3. 算术表达式、赋值运算符

## 3.1 变量

过年了,阿普拿到 m 元压岁钱,牛牛的压岁钱是阿普的 6 倍,牛牛的压岁钱是多少呢?显然,牛牛的压岁钱是 6m。这里的 m 是用字母表示的未知量,也就是程序里变量。

在程序中,数据是以变量和常量 2 种形式存在的。

变量由两个要素构成:**变量的名称和变量的类型**。变量的名称是这段存储空间是唯一标识。用来标识变量、符号常量、函数、数组、类型等实体名字的有效字符序列称为标识符。

C++规定标识符只能由字母、**数字和下划线 3 种字符组成,且第一个字符必须为字母或下划线**。而且,为了便于阅读和理解程序,变量名等标识符在命名是最好能“见名思义”,也就是根据变量名就能确定该变量的含义和作用。如:

```
int age, number; //分别表示年龄, 数值
double score, average, max, min; //分别表示: 分数、平均分、最高分、最低分
char a='A';//定义单个字符
bool flag=true;
```

C/C++提供了 12 种基本数据类型, 每种数据类型占用的字节数和表示范围各不一样。常用的数据类型只有以下几种。

**int:** 有符号的整型。其表示范围是 $-2^{31} \sim 2^{31}-1$ , 占 4 个字节,约 20 亿。

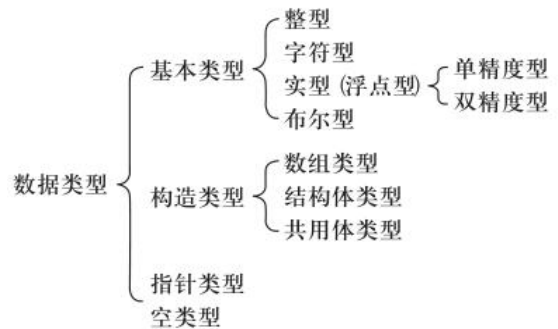
**double:** 双精度浮点型(即通常所说的实数), 占 8 个字节。其数值范围为 $10^{-307} \sim 10^{+308}$

**float:** 单精度浮点型(即通常所说的实数), 占 4 个字节,其数值范围为 $10^{-38} \sim 10^{+38}$

**char:** 字符类型, 占 1 个字节

**long long** 长整数类型, 其表示范围是 $-2^{63} \sim 2^{63}-1$ , 占 8 个字节,约  $10^{19}$  左右。

**bool** 布尔类型, 占 1 个字节, 表示 0 或 1, 或者 TRUE 或 FALSE。



## 3.2 常量

为了给变量赋值, 通常需要用常量。**所谓常量, 就是从字面上即可判别其值的量。**如“127”, “-456”, “3.1415926”, “-92.75”等。在 C/C++语言中, **常量有整型常量、浮点型常量、字符型常量等。**

### 1. 整型常量

一个整型常量可以赋值给整型变量、浮点型变量、甚至字符型变量。

```
int number = 17; //赋值后，整型变量 age 的值为 17
double area=29; //赋值后浮点型变量 area 的值为 29.0
```

## 2. 浮点型常量

一个浮点型常量可以赋值给浮点型变量、整型变量、甚至字符型变量。

```
int number = 17.23; //赋值后，整型变量 age 的值为 17
double area=29.75; //赋值后浮点型变量 area 的值为 29.75
```

## 3. 字符常量

字符常量是用单引号括起来的单个字符。例如，'a'、'b'、'='、'+','?' 等都是合法字符常量。字符常量的特点如下。

1) 字符常量只能用单引号括起来，不能用双引号或其他括号。单引号中的字符不能是单引号“'”和反斜杠“\”，它们特有的表示法在下一节的转义字符中介绍。

2) 字符常量只能是单个字符，不能是字符串，如'ab'就是错误的。

3) 字符可以是字符集中任意字符。

## 4. 字符串常量

字符串常量是由一对双引号括起的字符序列。例如，"CHINA"、"C program:"、"\$12.5"、"A"等都是合法的字符串常量。

字符串常量和字符常量是不同的量。它们之间主要有以下区别。

1) 字符常量由单引号括起来；字符串常量由双引号括起来。

2) 字符常量只能是单个字符；字符串常量则可以含一个或多个字符。

3) 一个字符常量可以赋予一个字符变量；字符串常量则通过一个字符型数组或字符型指针变量存储。

C++ 中定义一个符号常量的格式如下\*：

类型名 const 常量名；

或 const 类型名 常量名；

例如：

```
const int a = 100;
const double PI = 3.1415926;
```

使用符号常量是为了方便程序的修改，增强程序的读性。为了更清晰地区别常量和变量，常量名通常用大写字母表示。

## 3.3 小数点的设置

在结果输出时，常常碰到有精度要求的问题，即确定小数点位数。设置小数点位数的方法主要有 3 种：

方法 1：采用 `printf("%.xlf",a)` 实现。 “.” 语法格式，进行小数点设置；x 表示小数点位数，lf 表示 double 类型。

```
#include<iostream>
using namespace std;
int main()
{
    double a;
    float b;
    a=1.0/3;
```

```

b=1.0/6;
cout<<a<<" "<<b<<endl;
printf("%.6f\n",a);
printf("%.20lf\n",b);
}

```

方法 1:

方法 2:

1. 添加头文件 `iomanip`, `iomanip` 是 C++ 中代表输入输出格式设置的头文件。
2. 用 `setprecision(n)` 设置小数点的位数, `n` 表示小数点的位数。

```

#include<iostream>
#include<iomanip> //精度设置头文件
using namespace std;
int main()
{
    double a=3.14159265358;
    cout<<a<<endl; //默认 输出为 3.14159
    //加入 setprecision(n) 设置浮点数有效数字
    cout<<setprecision(3)<<a<<endl; //将精度改为 3 (即有效数字三位) 输出 3.14
    cout<<setprecision(10)<<a<<endl; //将精度改为 10 输出 3.141592654
    return 0;
}

```

方法 3:

1. 用 `cout` 的成员函数设置小数点位数: `cout.precision(n)`
2. 用 `fixed` 固定住小数点的位数。

```

#include<iostream>
using namespace std;
int main()
{
    double a=3.14159265358;
    cout.precision(3);
    cout<<fixed<<a<<endl; //将精度改为 3 (即有效数字三位) 输出 3.14
    cout.precision(10);
    cout<<fixed<<a<<endl; //将精度改为 10 输出 3.141592654
    return 0;
}

```

### 3.4 表达式

所谓表达式,就是通过一些运算符将一些变量、常量等连接起来的式子。运算符可分为算术运算符、关系运算符、逻辑运算符、赋值运算符、条件运算符等。

表 3-1 C++语言的运算符分类

运算符类型	运算符及含义	说 明
算术运算符	加(+)、减(-)、乘(*)、除(/)、求余(或称模运算, %)、自增(++)、自减(--)	用于各类数值运算
关系运算符	大于(>)、小于(<)、等于(==)、大于等于(>=)、小于等于(<=)、不等于(!=)	用于比较运算
逻辑运算符	逻辑与(&&)、或(  )、非(!)	用于逻辑运算
位运算符	位与(&)、位或( )、位非(~)、位异或(^)、左移(<<)、右移(>>)	参与运算的量, 按二进制位进行运算
赋值运算符	简单赋值(=) 复合算术赋值(+ =, - =, * =, / =, % =) 复合位运算赋值(& =,   =, ^ =, >> =, << =)	用于赋值运算
条件运算符	条件求值(? :)	三目运算符, 用于条件求值
逗号运算符	,	用于把若干表达式组合成一个表达式
指针运算符	取内容(*)、取地址(&)	用于取内容和取地址
字节数运算符	sizeof()	用于计算数据类型所占的字节数
特殊运算符	括号(), 下标[], 成员(->, .)	用于改变运算顺序、获得构造类中的成员

### 3.4.1 算术运算符

算术运算符用于数值运算。算术运算符包括以下 7 个：加(+)、减(-)、乘(\*)、除(/)、求余数(%)、自增(++)、自减(--)。

#### 1. 模运算符

求余数的运算符“%”也称为模运算符。它是双目运算符，即需要有两个操作数。并且两个操作数都必须是整数类型。

$a \% b$  的结果就是  $a$  除以  $b$  的余数。

另外，任意一个正整数 NUM 对正整数 N 取余，结果总是落在  $[0, 1, 2, \dots, N-1]$  的范围内

#### 2. 除法运算符

C/C++ 的除法运算符有个特殊之处，即：如果  $a$ 、 $b$  是两个整数类型的变量或者常量，那么  $a/b$  的结果是  $a$  除以  $b$  的商，不保留余数。

例如，表达式“5/2”的结果是 2，而不是 2.5，如果希望得到的结果保留余数，要使用表达式“ $1.0 * a/b$ ”。又如，表达式“ $1/5$ ”的结果是 0，而不是 0.2。

#### 3. 自增运算符

“++”用于将整型或浮点型变量的值加 1。只有一个操作数，是单目运算符；并且该操作数必须是变量，而不能是常量或表达式等。

自增运算符有两种用法：

用法 1：变量名++，i++，此时“++”称为“后置++”。

\*用法 2：++变量名，++i，此时“++”称为“前置++”。

这两种用法都会使得变量的值加 1，但它们是有区别的。

```
#include<iostream>
using namespace std;
int main()
{
```

```

int n1=3, n2;
n2 = n1++; // (1) “后置++”运算符，先自加，再赋值
cout<<"n1="<<n1<<" n2="<<n2<<endl;
n1= 3; // (3) 重新将 n1 赋值为 3
n2 = ++n1; // (2) “前置++”运算符，先赋值，再自加
cout<<"n1="<<n1<<" n2="<<n2<<endl;
return 0;
}

```

#### 4. 算术表达式

算术表达式就是通过算术运算符把变量、常量等连接起来的式子，如“ $b\%10 + 25$ ”、“ $a*3 - 4*c$ ”等都是算术表达式。算术表达式的值就是算术运算的结果。假设变量 a、b、c 的值分别为 27、5、2，则前面两个表达式的值分别为 32 和 7。

数值计算是所有高级语言的非常典型的应用之一。为了能让 C++ 语言程序进行数值计算，还必须将代数式写成 C++ 语言合法的表达式。

由算术运算符、括弧、内部函数及数据组成的式子称为算术表达式。

例如，下面数学表达式对应的 C++ 语言表达式。

$\frac{a+b}{a-b}$	$(a+b)/(a-b)$
$x^2 - e^5$	$x*x-\exp(5.0)$

### 3.4.2 赋值运算符

赋值运算符“=”用于对变量进行赋值。赋值的含义就是赋予变量一个新的值，替换原有的值。

赋值表达式的一般形式为：

变量=表达式

在赋值符“=”之前加上其他运算符，就构成了复合的运算符。如“+=”、“-=”、“\*=”、“/=”、“%=”等。

例如： $a+=3$  等效于  $a=a+3$     $a*=4*x+y$  等效于  $a=a*(4*x+y)$

**例 1** 没有赋值过的变量，其值是不确定的。

```

#include <iostream>
using namespace std;
int main()
{
    int a = 5, b;
    cout<<"a="<<a<<" b="<<b<<endl;
    return 0;
}

```

赋值时的自动类型转换：如果赋值时，赋值运算符“=”左右两边的数据类型不一样，就要进行自动类型转换。

`int number, age = 17.5;` //赋值后，整型变量 age 的值为 17



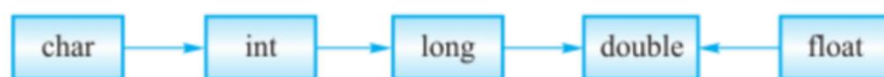
```
number = 3.14*3*3;           //赋值后整型变量 number 的值为 28
double area=number;         //赋值后浮点型变量 area 的值为 28.0
```

**例 2** 编写程序，实现交换程序中两个变量的值。

```
#include <iostream>
using namespace std;
int main( )
{
    int a, b;           //定义两个整型变量
    cin>>a>>b;         //从键盘上输入数据
    int t;             //用来保存变量 a 的值的临时变量
    //以下三条语句用于交换 a 和 b 的值
    t = a;             //(1)
    a = b;             //(2)
    b = t;             //(3)
    cout<<"a = "<<a<<" , b = "<<b;
    return 0;
}
```

## 3.5 类型转换

整型、实型、字符型数据之间可以进行混合运算。由于参与混合运算的各数据的类型不同，在运算时需要进行类型转换。数据类型转换有两种方式：隐式类型转换和强制类型转换。



### 1. 隐式数据类型转换

如图箭头表示当运算对象为不同类型时转换的方向，可以看到箭头由数据长度小的向数据长度大的转换，即按数据长度增加的方向进行，保证精度不降低。

例如，为了计算表达式  $3/2+1.5$  的值，首先作运算“ $3/2$ ”，两个整型数据相除，其结果仍为整型，计算值为 1。然后作运算“ $1+1.5$ ”，发现类型不一样，进行类型转换，将整型数 1 转换成实型数 1.0 后再运算，即  $1.0+1.5$ ，最后计算结果为 2.5。

### \*2. 强制类型转换

强制类型转换符是由类型符加一对圆括号构成，其功能是强制将一个表达式结果的数据类型转换为特定类型。

强制类型转换的一般形式为：

**(类型符)表达式 (double)x/2**

表示将变量 x 的内容强制转换为浮点数，再除以 2。

说明：

- 1) 类型符和表达式都需要加括号（单个变量可以不加括号）。
- 2) 无论隐式转换，强制转换都是临时转换，不改变数据本身的类型和值。

## 3.6 上课习题

### 1 [3230]成绩

牛牛最近学习了 C++ 信奥入门课程，这门课程的总成绩计算方法是：

总成绩=作业成绩×20%+小测成绩×30%+期末考试成绩×50%

牛牛想知道，这门课程自己最终能得到多少分（取整数）。

```
#include<bits/stdc++.h>
using namespace std;
int main ()
{
    int a,b,c;
    cin>>a>>b>>c;//输入 a,b,c;
    cout<<a*0.2+b*0.3+c*0.5<<endl;
}
```

### 2 [4019] 计算面积

从键盘输入一个梯形的上底、下底和高（整型数据），输出该梯形的面积（保留小数点后 2 位）。

方法 1:

```
#include<iostream>
using namespace std;
int main(){
    double a,b,c,sum;
    1
    2
    3
    cout<<fixed<<sum;
    return 0;
}
```

方法 2:

```
#include<iostream>
using namespace std;
int main(){
    double a,b,c,sum;
    cin>>a>>b>>c;
    sum=(a+b)*c/2;
    printf(_____);
    return 0;
}
```

### 3 [2160] 求 2 个数的和差积

输入 2 个整数 num1 和 num2，计算并输出它们的和、差、积、商与余数。

样例输入

5 3

样例输出

5+3=8

5-3=2

5\*3=15

5/3=1

5%3=2

```
#include<iostream>
```

```
#include<cstring>
```

```
using namespace std;
```

```
int main(){
```

```
    int a,b;
```

```
    cin>>a>>b;
```

```
    1
```

```
    2
```

```
    3
```

```
    4
```

```
    5
```

```
    return 0;
```

```
}
```

### 4 [2567] 计算分数的浮点数值

两个整数 a 和 b 分别作为分子和分母，既分数 a/b，求它的浮点数值(双精度浮点数，保留小数点后 9 位)。

输入

输入仅一行，包括两个整数 a 和 b。

输出

输出也仅一行，分数 a/b 的浮点数值（双精度浮点数，保留小数点后 9 位）。

样例输入

5 7

样例输出

0.714285714

```
#include <iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    1
```

```
    cin>>a>>b;
```

```

c=a/b;
cout.precision(9);//设置精度 cout.precision
2
return 0;
}

```

## 5 [2569] 计算多项式的值

对于多项式  $f(x)=ax^3+bx^2+cx+d$  和给定的  $a,b,c,d,x$ ，计算  $f(x)$  的值，保留到小数点后 7 位。

输入仅一行，包含 5 个实数，分别是  $x$ ，及参数  $a$ 、 $b$ 、 $c$ 、 $d$  的值，每个数都是绝对值不超过 100 的双精度浮点数。数与数之间以一个空格分开。

输出一个实数，即  $f(x)$  的值，保留到小数点后 7 位。

样例输入 2.31 1.2 2 2 3

样例输出 33.0838692

```

#include<iostream>
using namespace std;
int main(){
double x,a,b,c,d,sum;
1
2
cout.precision(7);
cout<<fixed<<sum<<endl;
return 0;
}

```

## 课堂作业列表

2323 黑鸡跑 1000	4019 求梯形面积
2861 输出对调后的数	3228 小 Q 买笔
1013 闰年判定	2595 苹果和虫子
2596 判断数正负	

## 习题

- C 语言中合法的字符常量是( )  
 A) n                      B) '\n'                      C) 110                      D) "n"
- 下面的程序段将输出( )  

```

double base = 2.5, heigh = 2, area;
area = 1 / 2 * base * heigh;
cout<<area<<endl;

```

 A.0                      B.2                      C.2.5                      D.3
- 下面的程序段将输出 ( )

```
int n = 10, s;
s = 1 / 2 * n * (n + 1);
cout<<s<<endl;
```

- A.0                    B.27.5                    C.55                    D.110

4. 下面的程序段将输出 (                    )

```
double r = 1, v;
v = 4 / 3 * 3.14 * r * r * r;
printf("%.2f\n", v);
```

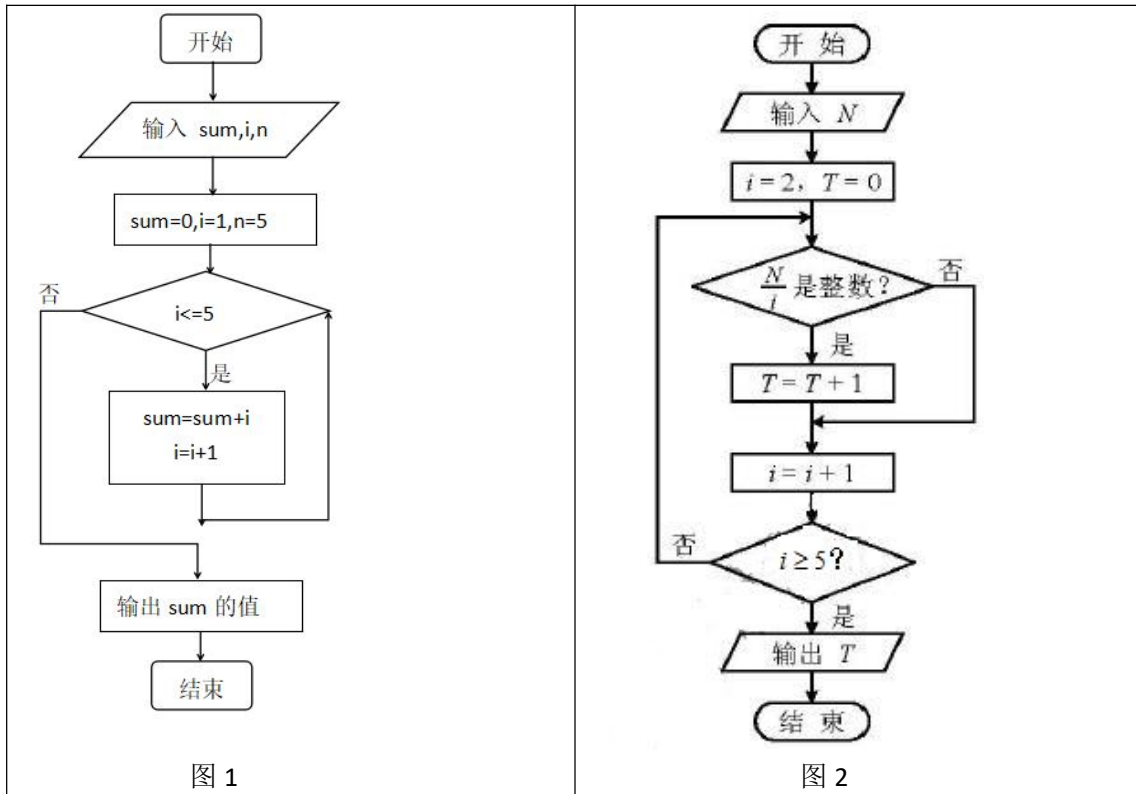
- A.0.00                    B.3.00                    C.3.14                    D.4.19

5. 阅读如图 1 程序框图, 输入的结果为(                    )

- A) 10                    B)15                    C)20                    D)25

6. 阅读如图 2 的程序框图, 运行相应的程序, 若输入  $N$  的值为 20, 则输出  $T$  的值为(                    )

- (A) 1                    (B) 2                    (C) 3                    (D) 4



7. 阅读程序写结果。

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"98+2=";
    cout<<100;
    return 0;
}
```

输出: \_\_\_\_\_

## 8. 变量定义

- (1) 定义一个 int 类型的变量 a \_\_\_\_\_  
将定义的变量 a 赋值为 8 \_\_\_\_\_
- (2) 定义一个 int 类型的变量 b \_\_\_\_\_  
将定义的变量 b 赋值为 10 \_\_\_\_\_

## 9. 阅读程序写结果

```
#include<iostream>
using namespace std;
int main() {
    int i, j, k;
    i=8;
    j=9;
    k=i+j;
    cout<<i;
    cout<<j;
    cout<<k;
    return 0;
}
```

请问 cout<<i; 执行后会在屏幕上显示\_\_\_\_\_

请问 cout<<j; 执行后会在屏幕上显示\_\_\_\_\_

请问 cout<<k; 执行后会在屏幕上显示\_\_\_\_\_

10. 已知某班有男同学 x 位，女同学 y 位，x 位男生平均分是 87 分，y 位女生的平均分是 85，问全体同学平均分是多少分

```
#include<iostream>
using namespace std;
int main()
{
    _____1_____
    cin>>x>>y; //输入男女人数
    cout<<_____<<endl;
}
```

11. 编写程序求矩形的面积和周长，矩形的长和宽由键盘输入，请填空。

```
#include <stdio.h>
void main()
{ double l,w;
    _____1_____
    cin>>l>>w;//scanf("%lf%lf",&l,&w);
    area=_____2_____;
    girth=_____3_____;
    _____4_____
}
```

```
}
```

12. 给定被除数和除数，求整数商及余数。此题中请使用默认的整除和取余运算，无需对结果进行任何特殊处理。

```
#include<iostream>
using namespace std;
int main(){
    int a,b;
    _____ 1 _____ //输入 a,b;
    cout<<_____ 2 _____<<" "<<_____ 3 _____ ;
}
```

13.

```
#include <iostream>
using namespace std;
int a,b;
int main()
{
    cin>>a>>b;
    a=a+b;
    b=a-b;
    a=a-b;
    cout<<a<<" "<<b<<endl;
    return 0;
}
```

输入：

11 99

输出： \_\_\_\_\_

# 第四讲 关系与逻辑

主要内容:

1. 掌握关系运算符及关系表达式
2. 掌握逻辑表达式

## 4.1 关系运算符及关系表达式

用关系运算符将两个表达式连接起来的式子，称为关系表达式。关系表达式的一般形式可以表示为：

表达式 关系运算符 表达式

**关系表达式的值只有两种：0 或者 1。**0 代表关系判断不成立，1 代表关系判断成立。

C++语言中的关系运算符的运算规则如表 4-1 所示。

表 4-1 C++语言关系运算符

运算符	含义	优先级	实例	结果
<	小于	比算术“+、-”运算低	15+10<20	0
<=	小于或等于		10<=20	1
>	大于		10>20	0
>=	大于或等于		32>=15+23	0
==	等于	高于位与运算(&)符	'T'=='t'	0
!=	不等于		'T'!='t'	1

强调：**== (等于) != (不等于)**

注意：在判断两个表达式是否相等时，一定要用两个等于号“==”，而不能用一个等于号“=”，后者表示赋值。

## 4.2 逻辑运算符

逻辑运算符有 3 种：**!(逻辑非)、&&(逻辑与)、||(逻辑或)**。逻辑表达式的一般形式为：

表达式 逻辑运算符 表达式

其中的表达式可以又是逻辑表达式，从而组成了嵌套的情形，如(a&&b)&&c，根据逻辑运算符的左结合性，也可写为：a&&b&&c。逻辑表达式的值是式中各种逻辑运算的最后值，其值为 True（真）或 False（假），True 表示表达式成立, False 表示表达式不成立，也以用 1 表示 True，0 表示 False（假）。

以“1”和“0”分别代表“真”和“假”，运算规则如图所示。



exp1	exp2	exp1&& exp2	exp1    exp2	! exp1
真	真	真	真	假
真	假	假	真	假
假	真	假	真	真
假	假	假	假	真

**逻辑型变量要用类型标识符 bool 来定义**，它的值只能是 true 和 false 之一。逻辑型常量只有两个，即 false(假)和 true(真)。

```
bool found, flag = false; //定义逻辑变量 found 和 flag，并使 flag 的初值为 false
found = true;           //将逻辑常量 true 赋给逻辑变量 found
```

注意：编译系统在处理逻辑型数据时，将 false 处理为 0，将 true 处理为 1，在内存中占一个字节，而不是将字符串“false”和“true”存放在内存中。

例如，如果 n 的值为 4，那么“n>4&& n<5”的值就是假；“n>=2 && n<5”的值就是真。逻辑“或”( || )运算。例子：如果 n 的值为 4，那么“n>4 || n<5”的值就是真；

例 1. 阅读程序，写出程序的运行结果。

```
#include<iostream>
using namespace std;
int main(){
    int a,b,c,d;
    a = 2;b = 3;c = 0;
    d = (a > b) && (c = 4);
    cout << d << " " << c << endl;
    return 0;
}
```

例 2 阅读程序，写出程序的运行结果。

```
#include<iostream>
using namespace std;
int main( )
{
    int a = 4, b = 5, c;
    c = a > b; //(1)把关系表达式"a > b"的值赋值给变量 c
    cout<<c<<endl;
    c = a < b; //(2)把关系表达式"a < b"的值赋值给变量 c
    cout<<c<<endl;
    c = a != b; //(3)把关系表达式"a != b"的值赋值给变量 c
    cout<<c<<endl;
    c = a == b; //(4)把关系表达式"a == b"的值赋值给变量 c
    cout<<c<<endl;
    c = a = b; //(5)把变量 b 的值赋值给 a，然后把 a 的值赋值给 c
    cout<<c<<endl;
    return 0;
}
```

输出的结果是：\_\_\_\_\_

### 4.3 运算符的优先级和结合性

优先级	运算符	结合性	优先级	运算符	结合性
1	() , []	从左到右	7	==, !=	从左向右
2	!, +(正), -(负), ~, ++, --	从左到右	8	&&	从左到右
3	*, /, %	从左到右	9		从左到右
4	+(加), -(减)	从左到右	10	?:	从右向左
5	<<, >>	从左到右	11	=, +=, *=, /=	从右向左
6	<, <=, >, >=	从左到右			

优先级和结合性是运算符的两个重要的特性。求解表达式时的顺序，按以下两个原则进行：

先按运算符的优先级别高低次序执行；如果在一个操作数两侧的运算符的优先级别相同，则按结合性中规定的“结合方向”进行运算。

表达式： $a+b*c$ ，先执行乘法运算“ $b*c$ ”，再执行加法，把  $a$  的值与乘法运算的结果加起来，这体现了运算符的优先级。

表达式： $a=b=5$ ，等效于： $a=(b=5)$ ，即先执行“ $b=5$ ”的赋值表达式，然后把该表达式的值(就是变量  $b$  的值)赋值给变量  $a$ 。这体现了运算符的结合性。如表所示，为常用运算符的优先级。

### 4.4 \*常用的数学函数

```
int abs( int x );           //求整型数据 x 的绝对值
double fabs( double x );  //求浮点型数据 x 的绝对值
double sqrt( double x );  //求 x 的平方根
double pow( double x, double y ); //求指数  $x^y$ 
double ceil(double x) //求不小于 x 的最小整数（上取整）
double floor(double x) //求不大于 x 的最小整数（下取整）
```

函数格式	含义	举例
int abs( int i)	返回整型参数 i 的绝对值	abs(-5)=5
double fabs( double x)	返回双精度参数 x 的绝对值	fabs(-2.5)=2.5
double ceil( double x)	返回不小于 x 的最小整数(上取整)	ceil(2.5)=3.0 ceil(-2.5)=-2.0
double floor( double x)	返回不大于 x 的最大整数(下取整)	floor(2.5)=2.0 floor(-2.5)=-3.0
double pow( double x, double y)	返回 x 的 y 次幂的值	pow(2,3)=8
double sqrt( double x)	返回 x 的平方根	sqrt(9)=3
double log( double x)	返回 $\ln(x)$ 的值(以 e 为底)	log(2.71) $\approx$ 1 log(8)/log(2)=3
double log10( double x)	返回 $\lg(x)$ 的值(以 10 为底)	log10(100)=2

以 `pow` 函数为例加以说明，`pow` 函数的原型是：

函数名是 `pow`，函数名后面圆括号内用逗号隔开的是两个 `double` 型的参数，即在调用 `pow` 函数时，需要带两个参数；函数名前面有 `double` 类型说明符，表示该函数执行完毕会返回一个 `double` 型的数据，即  $x^y$  的结果。

例如要对 2.5 开 3 次方根，即要求  $2.5^{1/3}$ ，可使用下面的代码：

```
double x = 2.5, y = 1.0/3, z; //注意不能写成 1/3, 否则 y 的值为 0
```

```
z = pow(x,y); //对 x 开 3 次方根, 并把结果赋值给变量 z
```

注意，初学者容易将函数调用写成如下的形式，这都是错误的：

```
z = double pow(x, y); 错误
```

```
z = double pow( double x, double y); 错误
```

例：阅读并上机调试程序，体会数学函数的使用。

```
#include<iostream>
#include<cmath>
using namespace std;
int main(){
    int x1=-2,x2=8;
    double y1,y2,y3,y4,y5;
    y1=abs(x1);
    y2=sqrt(x2);
    y3=pow(3,2);
    y4=ceil(3.145);
    y5=floor(3.145);
    cout<<y1<<" "<<y2<<" "<<y3<<" "<<y4<<" "<<y5;
    return 0;
}
```

## 4.5 上课习题

### 1[2568] 甲流疫情死亡率

题目描述

甲流并不可怕，在中国，它的死亡率并不是很高。请根据截止 2009 年 12 月 22 日各省报告的甲流确诊数和死亡数，计算甲流在各省的死亡率。

输入：输入仅一行，有两个整数，第一个为确诊数，第二个为死亡数。

输出：输出仅一行，甲流死亡率，以百分数形式输出，精确到小数点后 3 位。

样例输入：10433 60

样例输出：0.575%

```
#include <iostream>
using namespace std;
int main(){
```

```

double a,b,c;
cin>>a>>b;
c=_____//结果*100 获得%的数字
cout.precision(3);//%是字符，放在引号里面输出
cout<<fixed<<c<<"%"<<endl;
return 0;
}

```

## 2 [3545]计算各位和

题目描述

输入一个三位数，求解各位数字之和。如 123,1+2+3=6.

样例输入

234

样例输出

9

分析：如何获得个位、十位、百位的值。

个位：三位数对 10 取余数

十位：三位数除以 10，再对 10 取余数

百位：三位数除以 100

```
#include<iostream>
```

```
using namespace std;
```

```
int main(){
```

```
    int x,a,b,c;
```

```
    cin>>x;
```

```
    a=  1
```

```
    b=  2
```

```
    c=  3
```

```
    int sum=a+b+c;
```

```
    cout<<sum;
```

```
}
```

## 3 [2612] 判断闰年

判断某年是否是闰年。如果公元 a 年是闰年输出 Y，否则输出 N。

输入

输入只有一行，包含一个整数 a(0 < a < 3000)。

输出

一行，如果公元 a 年是闰年输出 Y，否则输出 N。

样例输入

2006

样例输出

N

方法 1:

```
#include<iostream>
using namespace std;
int main(){
    int year;
    cin>>year;
    int f1,f2,f3,f4;
    f1=year%4==0;//表示是赋值 把 year%4==0 结果赋值给 f1
    f2=year%100!=0;
    f3=year%400==0;
    f4=f1&&f2||f3;
    cout<<f4;

    return 0;
}
```

方法 2:

```
#include<iostream>
using namespace std;
int main()
{
    int y,flag;
    cin>>y;
    flag=_____
    cout<<flag<<endl;
    return 0;
}
```

#### 4 \*踢足球

明明、亮亮、强强三人在社区运动场上踢足球，不小心将王老师家的玻璃窗打碎了.当王老师问他们是谁打碎了玻璃窗时，明明说：“是亮亮打的.”亮亮说：“不是我打的.”强强也说：“不是我打的.”经调查知，他们三人中只有一个人讲了实话.请问到底是谁打碎了玻璃窗

1. 写出表达式

明明说：“是亮亮打的.”\_\_\_\_\_

亮亮说：“不是我打的.”\_\_\_\_\_

强强也说：“不是我打的.”\_\_\_\_\_

## 课堂作业列表

- |                |                 |
|----------------|-----------------|
| 2567 计算分数的浮点数值 | 2568 甲流疫情死亡率    |
| 3545 计算各位和     | 2570 温度表达转化     |
| 2576 浮点数向零舍入   | 2576 浮点数向零舍入    |
| 2612 判断闰年      | 2160 求 2 个数的和差积 |

## 习题

- 表达式 $(x\%3)$ 等价于表达式( )  
A. $x\%3 == 2$                       B. $x\%3 != 0$                       C. $x\%3 == 1$                       D. $x\%3 == 0$
- 假设  $x$ 、 $y$ 、 $z$  为整型变量，且  $x=2$ ， $y=3$ ， $z=10$ ，则下列表达式中值为 1 的是( )  
A)  $x \&\& y \mid\mid z$                       B)  $x > z$   
C)  $(!x \&\& y) \mid\mid (y > z)$                       D)  $x \&\& !z \mid\mid !(y \&\& z)$
- 为表示关系  $x \geq y \geq z$ ，应使用 C 语言表达式( )  
A. $(x >= y) \&\& (y >= z)$                       B. $(x >= y) \text{AND} (y >= z)$   
C. $(x >= y >= z)$                       D. $(x >= y) \& (y >= z)$
- C 程序中，正确表示" $10 < a < 20$  或  $a > 30$ "的条件表达式为( )  
A)  $(a > 10 \&\& a < 20) \&\& (a > 30)$                       B)  $(a > 10 \&\& a < 20) \mid\mid (a > 30)$   
C)  $(a > 10 \mid\mid a < 20) \mid\mid (a > 30)$                       D)  $(a > 10 \&\& a < 20) \mid\mid !(a < 30)$
- 表达式  $1/5+3\%4+4.5/5$  的值是( )  
A) 3.9                      B) 0.9                      C) 1.100000                      D) 1.85
- 将数学表达式  $\frac{ab}{c+df}$  改写为 C 语言表达式，正确的是( )。  
A)  $ab/(c+df)$                       B)  $a*b/c+d*f$                       C)  $a*b/(c+d*f)$                       D)  $(a*b)/(c+d)*f$
- C 语言中，合法的数值常量是( )  
A) 5E                      B) 3                      C) 18H                      D) 028
- 有以下程序：  

```
main()  
{  
    int x,y,z;  
    x=1;  
    y=5;  
    z=x%y;  
    z=z+1;  
    cout<<z<<endl;  
}
```

程序运行后的输出结果是( )  
A) 1                      B) 2                      C) 5                      D) 0
- 设  $a=5$ ， $b=6$ ， $c=8$ ， $d=7$ ， $x=2$ ， $y=2$ ，执行  $(x=a > b) \mid\mid (y=c > d)$  后  $y$  的值是( )。  
A) 1                      B) 2                      C) 0                      D) 8
- 设  $a$ 、 $b$  和  $c$  都是 `int` 型变量，且  $a=7$ ， $b=8$ ， $c=9$ ，则以下表达式中值为 0 的是( )。  
A)  $a \&\& b \mid\mid (b == c)$   
B)  $a <= b$   
C)  $a \&\& b \mid\mid c$

D) !a||b&&!c

11. 若有定义"double a=1.5,b=3.5,c=5.2;"，则表达式 (a>=b && c!=b) || (!a && c-b) 的计算结果为\_\_\_\_\_。

12. 运行以下程序后，输出结果是\_\_\_\_\_。

```
main( )
{   int a,b;
    a=a+b;
    cout<<a<<endl;
}
```

13. C++语言中，关系表达式及逻辑表达式的值为\_\_\_\_\_或\_\_\_\_\_。

14. 计算多项式  $y=ax^3+bx^2+cx+d$  和给定的 a,b,c,d,x，计算 y 的值，保留小数点 2 位。

```
#include<iostream>
using namespace std;
int main(){
    _____;//1 定义
    cin>>x>>a>>b>>c>>d;
    _____//2
    cout.precision(2);
    _____//3 输出
}
```

15. 计算各位和：输入一个三位数，求解各位数字之和。如 123,1+2+3=6.

样例输入

234

样例输出

9

```
#include<iostream>
using namespace std;
int main(){
    int x,a,b,c;
    cin>>x;
    a=_____ ;
    b=_____ ;
    c=_____ ;
    int sum=_____ ;
    cout<<sum;
}
```

16. 判断某年是否是闰年。如果公元 a 年是闰年输出 Y，否则输出 N。

```
#include<iostream>
using namespace std;
int main(){
```

```

int year;
cin>>year;
int f1,f2,f3,f4;
f1= 1 //能被 4 整除
f2= 2 //不能被 100 整除
f3= year%400==0;
f4= 3
cout<<f4;
return 0;}

```

17.

```

#include <iostream>
using namespace std;
int main()
{
    int a,b;
    a=3,b=7;
    a=a-b;
    b=b+a;
    a=b-a;
    cout<<"a"<<a<<endl;
    cout<<"b"<<b<<endl;
    return 0;
}

```

输出: \_\_\_\_\_

18.

```

#include <iostream>
using namespace std;
int main()
{
    int a,b,c,ans;
    cin>>a>>b;
    c=a+b;
    a=c-a;
    ans=a*b;
    cout<<"Ans="<<ans<<endl;
    return 0;
}

```

输入:

5 8

输出: \_\_\_\_\_



# 第五讲 IF 语句

重点:

1. 理解 if 语句及 if 的两种用法
2. 理解分支结构 if()...else

## 5.1 C/C++的语句

C++语句是 C++语言程序的最基本成分。C++语句必须由分号“;”结尾，哪怕只有一个分号也能构成一个空语句。在 C 语言中只有“可执行语句”，没有“非执行语句”。

从形式上分，C++语句分为以下几类。

### 1. 控制语句

控制语句用于控制程序的流程，以实现程序的各种结构方式。它们由特定的语句定义符组成。C++语言有 9 种控制语句，分为以下 3 类。

- 1) 条件判断语句：if 语句、switch 语句。
- 2) 循环执行语句：for 语句、do while 语句、while 语句。
- 3) 转向语句：break 语句、continue 语句、return 语句。

### 2. 表达式语句

表达式语句由表达式加上分号“;”组成。其一般形式为：

表达式;

执行表达式语句就是计算表达式的值。例如，

```
x=y+z;      // 赋值语句
y+z;        // 加法运算语句，但计算结果没有保留，无实际意义
i++;        // 自增 1 语句，i 值增 1
```

### 3. 空语句

只有一个分号的语句，它什么也不做。有时用来做被转向点或循环语句中的循环体（循环体是空语句，表示循环体什么也不做）。

例如，下面的循环体为空语句。

```
while(cin>>a);
```

本语句的功能是：只要从键盘输入的一个数，一直循环让用户重新输入。

### 4. 复合语句

把多个语句用花括号“{}”括起来组成的语句称为复合语句。在程序中，应把复合语句看成是单条语句，而不是多条语句。例如，下面就是一条复合语句。

```
{ x=y+z;
  a=b+c;
  cout<<x<<a;}
```

复合语句内的各条语句都必须以分号“;”结尾，在括号“}”外不需再加分号。

### 5. \*逗号表达式

在 C++ 语言中，逗号“，”不仅是一个分隔符，也是一种运算符，称为逗号运算符。其功能是把两个表达式连接起来组成一个表达式，称为逗号表达式。

其一般形式为：

表达式 1, 表达式 2, 表达式 3,..., 表达式 n

其求值过程是：由左到右依次求表达式 1, 表达式 2, 表达式 3, …, 表达式 n 的值，并以最后表达式 n 的值作为整个逗号表达式的值。

```
int a=2,b=4,c=6,x,y;
t=x,x=y,y=t;
int main()
{   int a=2,b=4,c=6,x,y;
    y=((x=a+b++),(b+c), x+c);
    printf("y=%d,x=%d",y,x);
}
```

程序运行结果如下 y=12,x=6

本例中，y 等于整个逗号表达式的值，也就是表达式 3 的值，x 是第一个表达式的值。

对于逗号表达式还要说明以下几点。

- 1) 逗号运算符优先级最低。
- 2) 程序中使用逗号表达式，通常是要分别求逗号表达式内各表达式的值，并不一定要求整个逗号表达式的值。
- 3) 并不是在所有出现逗号的地方都组成逗号表达式，如在变量声明中或函数参数表中的逗号只是用作各变量之间的间隔符。

## 5.2 单分支结构

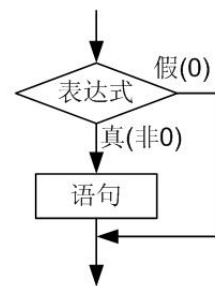
使用格式：

```
if(表达式)
    语句;
```

- 1) 语句的执行过程如图所示。
- 2) 表达式：一般为关系表达式、逻辑表达式，也可为算术表达式。其值按非 0 为 True，0 为 False 进行判断。
- 3) 如果当条件成立时要执行多个语句，就要使用“{ }”来构成复合语句。

例如，已知两个数 x 和 y，比较它们的大小，使得 x 大于等于 y。

```
if(x<y)
{t=x;x=y;y=t;} //t 为中间变量
```



(a) if 语句的第 1 种形式

## 5.3 双分支结构

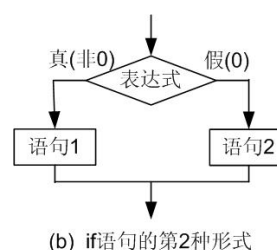
使用格式：

```
if(表达式)
    语句 1;
else
    语句 2;
```

语句的执行过程如图所示，即当表达式的值为非零（条件成立）时执行<语句 1>，否则执行 else 后面<语句 2>。

例如，输出 x 和 y 两个数中，值较大的一个。

```
if(x>y)
    cout<<x;
else
    cout<<y;
```



程序设计风格提示：从语法上来讲，整个 if 语句可以写在一行。如果条件表达式和语句都非常简单，那么整个语句可以写在一行；否则，最好在条件表达式后换行，而且语句部分要相对 if 缩进两格。

若条件成立时，要执行的操作由多个句子构成，我们必须把这些句子括在一对花括号{}内，我们称这种形式为语句块或复合语句。

```
if (条件表达式)
{
    语句 1;
    语句 2;
    .....
}
else
{
    语句 1;
    语句 2;
    .....
}
```

### 例 1 [2598]奇偶数判断

题目描述

给定一个整数，判断该数是奇数还是偶数。如果 n 是奇数，输出 odd；如果 n 是偶数，输出 even。

输入仅一行，一个大于零的正整数 n。

输出仅一行，如果 n 是奇数，输出 odd；如果 n 是偶数，输出 even。

```
#include<iostream>
using namespace std;
```

```

int main(){
int x;
cin>>x;
if(x%2==0)
    cout<<"even";
else
    cout<<"odd";
return 0;
}

```

## 例 2 [2191]2 月份的天数

从键盘上输入一个年份，输出该年 2 月份的天数。

```

#include <iostream>
using namespace std;
int main(){
    int n;
    cin>>n;
    if((n%100==0)&&(n%400==0) || (n%100!=0)&&(n%4==0))
        cout<<"29"<<endl;
    else
        cout<<"28"<<endl;
    return 0;
}

```

## 5.4 \*条件运算符和条件表达式

如果在条件语句中，只执行单个的赋值语句，常可使用条件表达式来实现。这样做不但使程序简洁，也提高了运行效率。

条件运算符为“?”和“:”，它是一个三目运算符，即有 3 个参与运算的量。由条件运算符组成条件表达式的一般形式为：

表达式 1? 表达式 2: 表达式 3

其求值规则为：先计算表达式 1，如果表达式 1 的值为真（即为非 0 值），则计算表达式 2，并以表达式 2 的值作为条件表达式的值，否则计算表达式 3，并以表达式 3 的值作为整个条件表达式的值。条件表达式通常用于赋值语句之中。

例如，条件语句是：

```

if(a>b) max=a;
else max=b;

```

可用条件表达式写为：max=(a>b)?a:b;

该语句的语义是：如 a>b 为真，则把 a 赋予 max，否则把 b 赋予 max。使用条件表达式时，还应注意以下几点。

1) 条件运算符的运算优先级低于关系运算符和算术运算符，但高于赋值符。因此，“max=(a>b)?a:b”可以去掉括号而写为“max=a>b?a:b”。

2) 条件运算符“?”和“:”是一对运算符,不能分开单独使用。

3) 条件运算符的结合方向是自右至左。

例如,“ $a>b?a:c>d?c:d$ ”应理解为: $a>b?a:(c>d?c:d)$ 。这也就是条件表达式嵌套的情形,即其中的表达式 3 又是一个条件表达式。

下面是用条件表达式对上例重新编程,输出两个数中的大数。

```
int main()
{
    int a,b,max;
    cin>>a>>b;
    cout<<"max="<<a>b?a:b;
}
```

## 5.5 案例讲解

### 1 [2601] 判断是否为两位数

判断一个正整数是否是两位数(即大于等于 10 且小于等于 99)。若该正整数是两位数,输出 1, 否则输出 0。

输入一个正整数, 不超过 1000。

输出一行。若该正整数是两位数, 输出 1, 否则输出 0。

样例输入 54

样例输出 1

```
#include<iostream>
using namespace std;
int main(){
    int x;
    cin>>x;
    if(_____)
        cout<<"1";
    else
        cout<<"0";
    return 0;
}
```

### 2[2603] 判断一个数能否同时被 3 和 5 整除

判断一个数  $n$  能否同时被 3 和 5 整除, 如果能同时被 3 和 5 整除输出 YES, 否则输出 NO。

输入输入一行, 包含一个整数  $n$ 。 (  $-1,000,000 < n < 1,000,000$  )

输出输出一行, 如果能同时被 3 和 5 整除输出 YES, 否则输出 NO。

样例输入 15

样例输出 YES

```

#include<iostream>
using namespace std;
int main(){
    int x;
    cin>>x;
    if(_____)
        cout<<"YES";
    else
        cout<<"NO";
    return 0;
}

```

### 3 [2600] 整数大小比较

输入两个整数，比较它们的大小。若  $x>y$ ，输出 $>$ ；若  $x=y$ ，输出 $=$ ；若  $x<y$ ，输出 $<$ 。  
 输入一行，包含两个整数  $x$  和  $y$ ，中间用单个空格隔开。 $0\leq x<232$ ， $-231\leq y<231$ 。  
 输出一个字符。若  $x>y$ ，输出  $>$ ；若  $x=y$ ，输出  $=$ ；若  $x<y$ ，输出  $<$ ；  
 样例输入 1000 100  
 样例输出  $>$

```

#include<iostream>
using namespace std;
int main(){
    int x,y;
    cin>>x>>y;
    if(_____)
        cout<<">";
    else {
        if(_____)
            cout<<"=";
        _____
            cout<<"<";
    }
    return 0;
}

```

### 4 [2608] 分段函数

题目描述

编写程序，计算下列分段函数  $y=f(x)$  的值。结果保留到小数点后三位。

$$y=-x+2.5; \quad 0\leq x<5$$

$$y=2-1.5(x-3)(x-3); \quad x \text{ 大于等于 } 5 \text{ 小于 } 10$$

$y=x/2-1.5; 10 \leq x < 20; 10 \leq x < 20$

输入一个数  $x$ ,  $0 \leq x < 20$ 。

输出  $x$  对应的分段函数值:  $f(x)$ 。结果保留到小数点后三位。

样例输入 1.0

样例输出 1.500

```
#include<iostream>
using namespace std;
int main()
{
    double x,y;
    cin>>x;
    if(_____)
        y=-x+2.5;
    else {
        if(_____)
            y=2-1.5*(x-3)*(x-3);
        else y=x/2-1.5;
    }
    cout.precision(3);
    cout<<fixed<<y<<endl;
    return 0;
}
```

## 课堂作业列表

- |      |                  |      |                     |
|------|------------------|------|---------------------|
| 2601 | 判断是否为两位数         | 2603 | 判断一个数能否同时被 3 和 5 整除 |
| 2604 | 判断能否被 3, 5, 7 整除 | 2607 | 骑车与走路               |
| 2608 | 分段函数             | 1045 | 分段函数                |

## 习题

### 一、选择题

- 表达式  $97 != 'a'$  的值是( )。  
A) true      B) false      C) 0      D) 1
- 能正确表示“ $10 \leq |x| \leq 99$ ”的表达式是( )。  
A)  $x \geq -99 \&\& x \leq -10 \&\& x \geq 10 \&\& x \leq 99$   
B)  $x > -99 \&\& x < -10 \mid \mid x >= 10 \&\& x <= 99$   
C)  $x \geq -99 \&\& x \leq -10 \mid \mid x \geq 10 \&\& x \leq 99$   
D)  $x > -99 \&\& x < -10 \&\& x >= 10 \&\& x <= 99$
- 判断 char 型变量 ch 是否为数字的表达式是( )。

- A) '0'<=ch<='9'  
 B) (ch>'0')&&(ch<'9')  
 C) (ch>='0')&&(ch<='9')  
 D) (ch>='0')||(ch<='9')
4. 下面的表达式中值为 0 的是( )。  
 A)3%5                      B)3/5.0                      C)3>5                      D)3<5
5. 与 x>0 不等价的表达式是( )。  
 A) !(x==0||x<0)  
 B) !(x==0)&&(x<0)  
 C) !(x<0)&&(x!=0)  
 D) !(x==0)&&!(x<0)
6. 能表示 a 不能被 2 整除且 a、b 不相等,但 a、b 的和等于 0 的 C 语言逻辑表达式是( )。  
 A) a==-b&&a%2==0  
 B) a!=b&&a+b==0&&a%2  
 C) !(a%2)&&a==-b  
 D) a%2==0&&a+b==0
7. 设 a=5,b=6,c=8,d=7,x=2,y=2,执行(x=a>b)|| (y=c>d)后 y 的值是( )。  
 A)1                      B)2                      C)0                      D)8
8. 设 a、b 和 c 都是 int 型变量,且 a=7,b=8,c=9,则以下表达式中值为 0 的是( )。  
 A) a&&b||(b==c)  
 B) a<=b  
 C) a&&b||c  
 D) !a||b&&!c
9. 在嵌套使用 if 语句时, C++ 语言规定 else 总是( )。  
 A. 和之前与其具有相同缩进位置的 if 配对                      B. 和之前与其最近的 if 配对  
 C. 和之前与其最近的且不带 else 的 if 配对                      D. 和之前的第一个 if 配对

10. 判断一个正整数是否是两位数(即大于等于 10 且小于等于 99)。若该正整数是两位数,输出 1, 否则输出 0。

```
#include<iostream>
using namespacestd;
int main(){
    int n;
    cin>>n;
    if(_____)cout<<1;
    else _____
    return 0;
}
```

11. 给定一个整数 N, 判断其正负。如果 N>0, 输出 positive; 如果 N=0, 输出 zero; 如果 N<0, 输出 negative。

```
#include<iostream>
using namespace std;
int main()
{
    int a;
```



```

    ___1_____;
    if(___2_____)
        cout<<"positive";
    else
    {
        if(___3_____)
            cout<<"zero";
        ___4____
            cout<<"negative";
    }
}

```

12. 写出下面程序的运行结果\_\_\_\_\_

```

#include<iostream>
using namespace std;
int main()
{
    int x=1,y=1,z=10;
    if(z<0)
    if(y>0) x=3;
    else x=5;
    cout<<x<<endl;
    if(z=y<0) x=3;
    else if(y==0) x=5;
    else x=7;
    cout<<x<<endl;
    cout<<z<<endl;
}

```

13.

```

#include <iostream>
using namespace std;
int main()
{
    int a,b,c,ans;
    cin>>a>>b;
    c=a+b;
    a=c-a;
    ans=a*b;
    cout<<"Ans="<<ans<<endl;
    return 0;
}

```

输入:

5 8

输出: \_\_\_\_\_

# 第六讲 多路选择

重点: if 多路选择

## 6.1 if 多路选择

语法格式:

```
if(表达式 1) 语句 1  
else if(表达式 2) 语句 2  
else if(表达式 3) 语句 3  
...  
else if(表达式 m) 语句 m  
else 语句 n
```

### 例 1: 分段函数

题目描述

编写程序, 计算下列分段函数  $y=f(x)$  的值。结果保留到小数点后三位。

```
y=-x+2.5; 0≤x<5  
y=2-1.5(x-3)(x-3); x 大于等于 5 小于 10  
y=x/2-1.5; 10≤x<20; 10≤x<20
```

输入

一个数  $x$ ,  $0 \leq x < 20$ 。

输出

输出  $x$  对应的分段函数值:  $f(x)$ 。结果保留到小数点后三位。

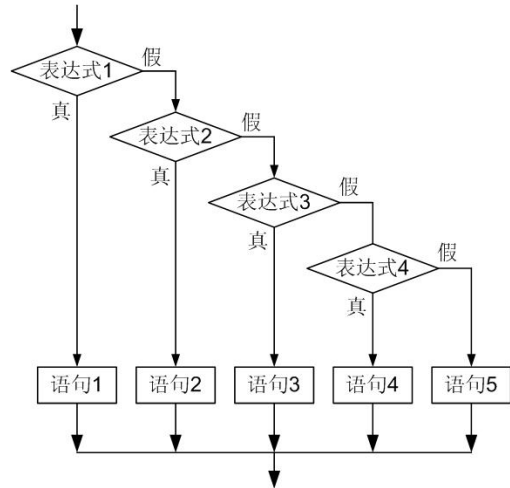
样例输入

1.0

样例输出

1.500

```
#include<iostream>  
#include <iomanip>  
using namespace std;  
int main()  
{  
    double x,y;  
    cin>>x;  
    if(0<=x&&x<5)  
        y=-x+2.5;  
    else if(5<=x&&x<10)  
        y=2-1.5*(x-3)*(x-3);  
    else y=x/2-1.5;
```



(c) if 语句的第3种形式

```

    cout<<fixed<<setprecision(3)<<y<<endl;
    return 0;

}

```

## 6.2 案例分析

### 1[2607] 骑车与走路

在清华校园里，没有自行车，上课办事会很不方便。但实际上，并非去办任何事情都是骑车快，因为骑车总要找车、开锁、停车、锁车等，这要耽误一些时间。假设找到自行车，开锁并车上自行车的时间为 27 秒；停车锁车的时间为 23 秒；步行每秒行走 1.2 米，骑车每秒行走 3.0 米。请判断走不同的距离去办事，是骑车快还是走路快。如果骑车快，输出一行"Bike"；如果走路快，输出一行"Walk"；如果一样快，输出一行"All"。

输入 输入一行，包含一个整数，表示一次办事要行走的距离，单位为米。

输出 输出一行，如果骑车快，输出一行"Bike"；如果走路快，输出一行"Walk"；如果一样快，输出一行"All"。

样例输入 120

样例输出 Bike

```

#include<iostream>
using namespace std;
int main(){
    double a,b,c;
    cin>>a;
    b=50+a/3;
    c=a/1.2;
    if ( _____ )
        cout<<"Walk"<<endl;
    else if ( _____ )
        cout<<"All"<<endl;
    else
        cout<<"Bike"<<endl;
    return 0;
}

```

### 2 [2614] 简单的计算器

一个最简单的计算器，支持+、-、\*、/ 四种运算。仅需考虑输入输出为整数的情况，数据和运算结果不会超过 int 表示的范围。然而：

1. 如果出现除数为 0 的情况，则输出：Divided by zero!
2. 如果出现无效的操作符(即不为 +、-、\*、/ 之一)，则输出：Invalid operator!

写法 1:

```
#include<iostream>
#include<cmath>
using namespace std;
int main()
{
    int a,b;
    char c;
    cin>>a>>b>>c;
    if(_____)
    cout<<"Invalid operator!";
    else if(_____)
    cout<<"Divided by zero!";
    else if(_____)cout<<a+b;
    else if(_____)cout<<a-b;
    else if(_____)cout<<a*b;
    else if(_____)cout<<a/b;
    return 0;
}
```

### 3 [1430] 求输出成绩等级

题目描述

给出一百分制成绩, 要求输出成绩等级'A'、'B'、'C'、'D'、'E'。90 分以上为 A 80-89 分为 B 70-79 分为 C 60-69 分为 D 60 分以下为 E

输入一个整数 0—100 以内

输出一个字符, 表示成绩等级

样例输入 90

样例输出 A

```
#include<iostream>
using namespace std;
int main(){
    int a;
    char s;
    cin>>a;
    if(1_____)s='A';
    else if(2_____)s='B';
    else if(3_____)s='C';
    else if(4_____)s='D';
    else s='E';
}
```

```
cout<<s;
return 0;
}
```

## 4 [2286] 五级成绩区间

输入五级成绩（A~E），输出相应的百分制成绩区间（0~100），要求使用多路选择语句。五级制成绩对应的百分制成绩区间为：A（90-100）、B（80-89）、C（70-79）、D（60-69）和 E（0-59）。

输入 B

输出 80-89

样例输入 E

样例输出 0-59

```
#include <iostream>
using namespace std;
int main()
{
    char a;
    cin>>a;
    if(_____)cout<<"90-100";
    else if(_____)cout<<"80-89";
    else if(_____)cout<<"70-79";
    else if(a=='D')_____
    else _____
    return 0;
}
```

## 课堂作业列表

2614 简单计算器	1014 计算运费
2608 分段函数	1045 分段函数
2945 星期几	1014 计算运费

## 习题

### 一、选择题

1. 关于 if 后面一对圆括号中的表达式，叙述正确的是( )。  
A) 只能用关系表达式  
B) 只能用逻辑表达式  
C) 只能用关系表达式或逻辑表达式

D) 可以使用任意合法的表达式

2. 执行以下程序段后, c 的值是( )。

```
int a=1,b=2,c=3;
if(a>=b)
    if(a==b) c=0;
    else c=1;
```

A)3                      B)1                      C)0                      D)2

3. 有以下程序:

```
main()
{   int m;
    m=-2;
    if(m=1)
        cout<<"@@@";
    else
        cout<<"%%%";
}
```

程序运行后的输出结果是( )。

A) @@@  
B) %%%  
C) %%  
D) @@@%%%

4. 有以下程序:

```
main()
{
    int a,b,t;
    cin>>a>>b;
    t=a;
    if(a<b)
        t=b;
    t=t*t;
    cout<<t<<endl;
}
```

程序运行后输入 3 和 4, 则程序的输出结果是( )。

A)9                      B)14                      C)16                      D)18

5. 有以下程序:

```
main()
{
    int a=5,b=0,c=0;
    if(a==b+c)
        cout<<"***";
        cout<<"####";
    else
        printf("####");
}
```

程序运行后的结果是( )。

- A) 有语法错误不能通过编译
- B) 输出\*\*\*\*####
- C) 可以通过编译，但是不能通过连接，因而不能运行
- D) 输出####

6. 有以下程序：

```
main()
{   int t;
    cin>>t;
    if(t>45) cout<<t;
    else cout<<t;
    if(t<35) cout<<t;
    else cout<<t;
    if(t>25) cout<<t;
}
```

程序运行后输入 65<回车>，程序的输出结果( )

- A)65
- B)6565
- C)656565
- D) 不确定的值

7. 有以下程序：

```
main( )
{   int a=2,b=4,c;
    if(a>b)
        c=1;
    else if(a==b)
        c=0;
    else
        c=-1
    printf(“%d”,c);
}
```

程序运行后的输出结果是( )

- A)-1
- B)0
- C)1
- D)2

8. 读程序写结果[noip2003]\_\_\_\_\_

```
#include <iostream>
#include <cstdio>
using namespace std;
int a,x,y,ok1,ok2;
int main()
{
    a=100;
    x=20;
    y=20;
    ok1=5;
    ok2=0;
    if((x>y) || ((y!=20)&&(ok1==0))&&(ok2!=0))
```

```

        a=1;
    else if((ok1!=0)&&(ok2==0))
        a=-1;
    else
        a=0;
    cout<<a<<endl;
    return 0;
}

```

### 9.简单计算器

```

#include<iostream>
using namespace std;
int main()
{
    int a,b;
    _____1_____
    cin>>a>>b>>ch;
    if(ch!='/'&&ch!='+'&&ch!='-'&&ch!='*') cout<<"Invalid operator!"<<endl;
    else {
        if(_____2_____)
            cout<<"Divided by zero!"<<endl;
        else {
            switch(_____3_____){
                case '+': cout<<a+b<<endl;
                case '-': cout<<a-b<<endl;
                case '*': cout<<a*b<<endl;
                _____4_____: cout<<a/b<<endl;
            }
        }
    }
    return 0;
}

```

### 10.

```

#include <iostream>
#include <string>
using namespace std;
int main()
{
    int a, b, c;
    a = 4, b = 5, c = 6;
    a = b + c;
    b = b - a;
    a = a - b;
    if (a < 0)

```



```

        if (b > 0)
            c = b + a;
        else
            c = b - a;
    if (c > 0)
        cout << "Wish you success!" << endl;
    else
        cout << "Good luck to you!" << endl;
    return 0;
}

```

输出: **Wish you success!**

输出: \_\_\_\_\_

11.

```

#include <iostream>
using namespace std;
int main()
{
    int a,b;
    cin>>a>>b;
    if(a>b)
        a=(a+b)/(a-b);
    else
        a=(a-b)%3;
    cout<<a<<endl;
    return 0;
}

```

输入 1:

24 12

输出 1: \_\_\_\_\_

输入 2:

7 12

输出 2: \_\_\_\_\_

# 第七讲 switch 语句

## 7.1 switch 语句

if 语句能方便处理从两者间选择之一，当要实现几种可能之一时，就要用 if...else if 甚至多重的嵌套 if 来实现。当分支较多时，程序变得复杂冗长，可读性降低，如例 4-5，虽然可以使用 if 语句来实现，但不直观。C 语言中提供了处理多分支的开关语句——switch 语句，可以方便、直观地处理多分支的控制结构，使程序变得简洁。

switch 语句的一般格式为：

```
switch(表达式)
{ case 常量表达式 1: 语句组 1;
  case 常量表达式 2: 语句组 2;
  ...
  case 常量表达式 n:语句组 n;
  default:语句组 n+1;
}
```

switch 语句的执行过程是：首先计算表达式的值，并逐个与其后的常量表达式值相比较，当表达式的值与某个常量表达式的值相等时，即执行其后的语句，然后不再进行判断，继续执行后面所有 case 后的语句，除非执行到 break 语句。如表达式的值与所有 case 后的常量表达式均不相同，则执行 default 后的语句。如果每个 case 分支的语句组中最后有 1 个 break 语句

在实际使用 switch 语句时，通常要求当执行完某个 case 后的一组语句序列后，就结束整个语句的执行，而不让它继续执行下一个 case 语句后面的语句序列，为此，可通过使用 break 语句来实现。该语句只有保留字 break，而没有其它任何成分。它是一条跳转语句，在 switch 中执行到它时，将结束该 switch 语句，系统接着向下执行其它语句。

```
switch( credit )
{
  case 1:  discount = 0.01;  break;
  case 2:  discount = 0.015; break;
  case 3:  discount = 0.022; break;
  case 4:  discount = 0.03;  break;
  case 5:  discount = 0.04;  break;
}
```

## 7.2 案例分析

### 1 [2945]星期几

根据从键盘上输入的代表星期几的数字，对应输出它的英文名称。

输入输入表示星期几的数字,其他情况输出 input error!

输出输出它的英文名称

1 Monday  
2 Tuesday  
3 Wednesday  
4 Thursday  
5 Friday  
6 Saturday  
7 Sunday

样例输入 1

样例输出 Monday

```
#include<iostream>
using namespace std;
int main()
{
    int weekday;
    cin>>weekday;
    switch(weekday)
    {
        case 1:cout<<"Monday"<<endl; break;
        case 2: cout<<"Tuesday"<<endl; break;
        case 3: cout<<"Wednesday"<<endl; break;
        case 4: cout<<"Thursday"<<endl; break;
        case 5: cout<<"Friday"<<endl; break;
        case 6: cout<<"Saturday"<<endl; break;
        case 7: cout<<"Sunday"<<endl; break;
        default:cout<<"input error!";
    }
    return 0;
}
```

## 2 [2614] 简单的计算器

一个最简单的计算器，支持+, -, \*, / 四种运算。仅需考虑输入输出为整数的情况，数据和运算结果不会超过 int 表示的范围。然而：

1. 如果出现除数为 0 的情况，则输出：Divided by zero!
2. 如果出现无效的操作符(即不为 +, -, \*, / 之一)，则输出：Invalid operator!

```
#include<iostream>
using namespace std;
int main()
{
    int a,b;
    char ch;
```

```

cin>>a>>b>>ch;
if(ch!='/'&&ch!='+'&&ch!='-'&&ch!='*')
    cout<<"Invalid operator!"<<endl;
else {
    if(ch=='/'&&b==0) cout<<"Divided by zero!"<<endl;
    else {
        switch(ch){
            case '+': cout<<a+b<<endl;
            case '-': cout<<a-b<<endl;
            case '*': cout<<a*b<<endl;
            defalut: cout<<a/b<<endl;
        }
    }
}
return 0;
}

```

### 3 [2286] 五级成绩区间

输入五级成绩（A~E），输出相应的百分制成绩区间（0~100），要求使用多路选择语句。五级制成绩对应的百分制成绩区间为：A（90-100）、B（80-89）、C（70-79）、D（60-69）和 E（0-59）。

输入 B

输出 80-89

样例输入 E

样例输出 0-59

```

#include <iostream>
using namespace std;
int main()
{
    char s;
    cin>>s;
    switch(_____){
        case 'A':cout<<"90-100";break;
        case 'B':cout<<"80-89";break;
        _____
        _____
        _____
    }
    return 0;
}

```

## 4 [1432] 算奖金

企业发放的奖金根据利润提成。利润低于或等于 100000 元的，奖金可提 10%；  
利润高于 100000 元，低于 200000 元 ( $100000 < I \leq 200000$ ) 时，低于 100000 元的部分按 10% 提成，高于 100000 元的部分，可提成 7.5%；  
 $200000 < I \leq 400000$  时，低于 200000 元部分仍按上述办法提成，（下同），高于 200000 元的部分按 5% 提成；  
 $400000 < I \leq 600000$  元时，高于 400000 元的部分按 3% 提成； $600000 < I \leq 1000000$  时，高于 600000 元的部分按 1.5% 提成；  
 $I > 1000000$  时，超过 1000000 元的部分按 1% 提成。从键盘输入当月利润 I，求应发奖金总数。  
输入一个整数，当月利润。  
输出一个整数，奖金。  
样例输入 900  
样例输出 90

```
#include<iostream>
using namespace std;
int main(){
    int n,m;
    cin>>n;
    int d=n/100000;
    if(d>10) d=10;
    switch(d){
        case 0: m=n*0.1; break;
        case 1: m=10000+0.075*(n-100000); break;
        case 2:
        case 3: m=17500+0.05*(n-200000); break;
        case 4:
        case 5: m=27500+0.03*(n-400000); break;
        case 6:
        case 7:
        case 8:
        case 9: m=33500+0.015*(n-600000); break;
        case 10: m=39500+0.01*(n-1000000); break;
    }
    cout<<m<<endl;
    return 0;
}
```

## 习题

### 一、选择题

1. 对 switch 后括号内的表达式，叙述正确的是( )

- A) 只能是数字
- B) 可以是浮点数
- C) 只能是整型数据或字符型数据
- D) 以上叙述都不对

2. 有如下程序

```
int main( )
{
    int x=1,a=0,b=0;
    switch(x)
    {
        case 0: b++;
        case 1: a++;
        case 2: a++;b++;
    }
    cout<<"a="<<a<<"b="<<b<<endl;
}
```

该程序的输出结果是\_\_\_\_\_。

- A. a=2,b=1
  - B.a=1,b=1
  - C.a=1,b=0
  - D.a=2,b=2
3. 若有 int x=67;，执行下列程序段后，变量 x 的正确结果是 ( )。

```
switch (x/10)
{
    case 7:
    case 6: cout<<"A";
    case 5: cout<<"B";break;
    default: cout<<"C";
}
```

- A. ABC
- B.A
- C.AB
- D.C

4. 有以下程序：

```
main( )
{
    int m;
    cin>>m;
    switch(m%4)
    {
        case 0:
            case 1:cout<<m;break;
            case 2:cout<<m*m;
            case 3:cout<<m*m*m;break;
    }
}
```

程序运行后输入 2<回车>, 程序的输出结果是( )  
 A)4                      B)9                      C)9 27                      D)4 8

5. 分段函数: 输入 x, 计算 y 值, 输出 y, 其中:

```

x<0 y=2x+3
x=0,y=0
x>0,y=(x+7)/3
int main()
{
    int x,y;
    scanf("%d",&x);
    if(x<0) _____ (1) _____ ;
    _____ (2) _____
    _____ (3) _____
    printf("%d",y);
}
    
```

6. 投票表决器: 输入 Y、y, 打印 agree; 输入 N、n, 打印 disagree; 输入其他, 打印 lose

```

main()
{
    char c;
    scanf("%c",&c);
    _____ (1) _____
    {
        case 'Y':
        case 'y': cout<<"agree"; _____ (2) _____;
        case 'N':
        case 'n': cout<<"disagree"; _____ (3) _____ ;
        _____ (4) _____: cout<<"lose";
    }
}
    
```

7. 一个最简单的计算器, 支持+, -, \*, / 四种运算。仅需考虑输入输出为整数的情况, 数据和运算结果不会超过 int 表示的范围。然而: 如果出现除数为 0 的情况, 则输出: Divided by zero! 如果出现无效的操作符(即不为 +, -, \*, / 之一), 则输出: Invalid operator!

```

#include<iostream>
using namespace std;
int main()
{
    int a,b;
    _1_____;//定义
    cin>>a>>b>>op;
    if(_2_____ ) cout<<"Divided by zero!"; //1 除 0 的情况
    else if(_3_____')
        cout<<"Invalid operator!";
    _4_____ {
        if(op=='+')cout<<a+b;
        else if(op=='-')cout<<a-b;
    }
}
    
```

```

        else if(op=='*')cout<<a*b;
        else cout<<a/b;
    }
    return 0;
}

```

8. 读程序写结果\_\_\_\_\_

```

#include <stdio.h>
int main()
{
    int x=1,y=1,z=10;
    if(z<0)
        if(y>0) x=3;
        else x=5;
        cout<<x<<endl;
    if(z=y<0) x=3;
    else if(y==0) x=5;
    else x=7;
    cout<<x<<endl;
    cout<<t<<endl;
}

```

9.

```

#include <iostream>
using namespace std;
int main()
{
    int n, i, a, b, c, d;
    cin>>a>>b>>c>>d;
    if (a>b)
        a=a/10;
    if (d>c)
        cout<<"TimeLimitExceeded";
    else if (a<b)
        cout<<"WrongAnswer";
    else if (a==b)
        cout<<"Accepted";
    return 0;
}

```

输入 1: \_\_\_\_\_

1000 100 765 155

输出 1

输入 2:

20 37 8 9

输出 2: \_\_\_\_\_



# 第八讲 推理

重点：掌握关系表达式表

## 1 [7599] 破案

某地发生了一件谋杀案，警察通过排查确定杀人凶手必为 4 个嫌疑犯中的一个。以下为 4 个人的供词：

A 说：不是我。

B 说：是 C。

C 说：是 D。

D 说：C 在胡说。

已知 3 个人说了真话，1 个人说了假话，根据这个信息写一个程序来确定到底谁是凶手。

输入输入一个字符，表示你认为的凶手

输出如果是凶手输出 yes, 否则输出 no.

样例输入 A

样例输出 no

分析：

### 1. 怎么表示 A,B,C,D?

killer=1 表示 A, killer=2, 表示 B

killer=3 表示 C, killer=4, 表示 D

### 2. 这么表示供词?

A 说：不是我。      killer!=1

B 说：是 C。          killer==3

C 说：是 D。          killer==4

D 说：C 在胡说。      killer!=4

### 3. 已知 3 个人说了真话，1 个人说了假话

```
sum=(killer!=1)+(killer==3)+(killer==4)+(killer!=4);
```

```
if(sum==3)
```

```
{
```

```
    cout<<killer;
```

```
}
```

### 4. 根据这个信息写一个程序来确定到底谁是凶手。

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main(){
```

```
    char k;
```

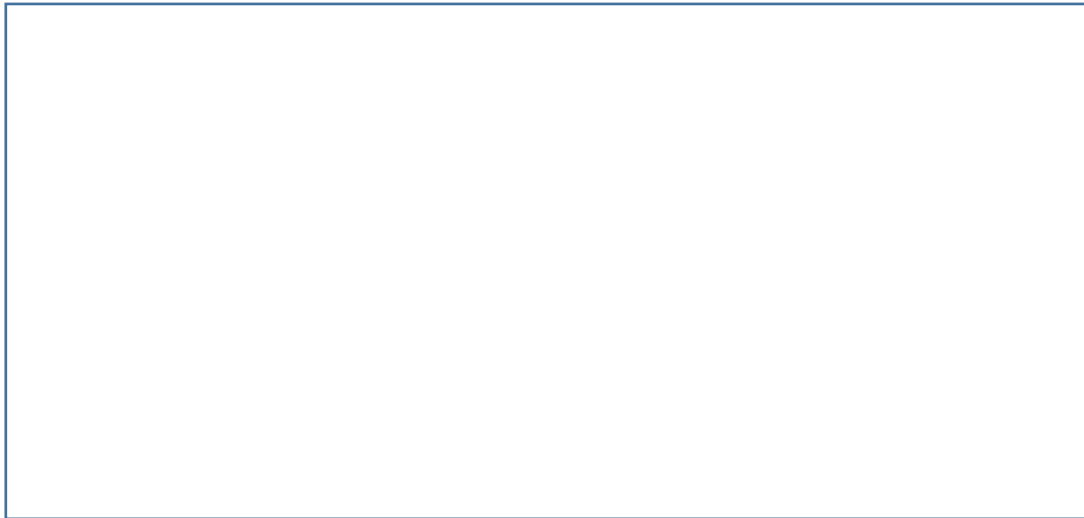
```
    int killer,sum=0;;
```

```

cin>>k;
killer=k-'A'+1;
sum=(killer!=1)+(killer==3)+(killer==4)+(killer!=4);
if(sum==3)
    cout<<"yes";
else
    cout<<"no";
return 0;
}

```

思考：如果用字符直接表示 A,B,C,D 要怎么表示？



### 随堂练习 1

明明、亮亮、强强三人在社区运动场上踢足球，不小心将王老师家的玻璃窗打碎了。当王老师问他们是谁打碎了玻璃窗时，明明说：“是亮亮打的。”亮亮说：“不是我打的。”强强也说：“不是我打的。”经调查知，他们三人中只有一个人讲了实话。请问到底是谁打碎了玻璃窗

#### 2. 写出表达式

明明说：“是亮亮打的。”\_\_\_\_\_

亮亮说：“不是我打的。”\_\_\_\_\_

强强也说：“不是我打的。”\_\_\_\_\_

### 2 [7853] 围棋赛 2

A、B、C、D、E 五人参加围棋赛，四位观战者预测了结果。甲说：“E 第 3，A 第 4。”乙说：“A 第 3，B 第 1。”丙说：“B 第 4，E 第 2。”丁说：“D 第 1，C 第 3。”实际结果是每人只猜对了-一个。参赛五人没有并列名次。

输入

A、B、C、D、E 五人的排名

输出

如果该排名是正确的，输出 yes，错误的输出 no

样例输入

1 2 3 4 5

样例输出

no

1. 写出表达式

- 甲说：“E 第 3，A 第 4。” \_\_\_\_\_
- 乙说：“A 第 3，B 第 1。” \_\_\_\_\_
- 丙说：“B 第 4，E 第 2。” \_\_\_\_\_
- 丁说：“D 第 1，C 第 3。” \_\_\_\_\_

2. 每人只猜对了一个

---

3. 甲乙丙丁的值不能相等，如何表达？

---

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main () {
```

```
    int a,b,c,d,e;
```

```
    int jia,yi,bin,ding;
```

```
    cin>>a>>b>>c>>d>>e;
```

```
        jia=(e==3)+(a==4);//甲乙丙丁的四句话
```

```
        yi=(a==3)+(b==1);
```

```
        bin=(b==4)+(e==2);
```

```
        ding=(d==1)+(c==3);
```

```
        if( jia==1&&yi==1&&bin==1&&ding==1){//一句话对
```

```
            cout<<"yes";
```

```
        }
```

```
        else
```

```
            cout<<"no";
```

```
    return 0;
```

```
}
```

### 3[7628] 数学竞赛

小张、小王、小李、小赵同时参加一次数学竞赛，赛后，小张说：“小李得第 x 名，我得第三名。”

小王说：“我得第一名，小赵得第四名。”

小李说：“小赵得第  $y$  名，我得第三名。”

小赵没有说话.成绩揭晓时，发现他们每个人的话都只说对了一半。  
问，他们四个人的名次到底是怎样的。

输入  $x,y$

输出小张、小王、小李、小赵的名次。

样例输入 1 2

样例输出 3 1 4 2

**思考：**

写出表达式

小张说：“小李得第  $x$  名，我得第三名。” \_\_\_\_\_

小王说：“我得第一名，小赵得第四名。” \_\_\_\_\_

小李说：“小赵得第  $y$  名，我得第三名。” \_\_\_\_\_

#### 4 [7629] 苹果数量\*

刘红、陈明、李小明三人各有一些苹果.

刘红说：“我有  $a$  个苹果，比陈明少 2 个，比李小明多 1 个。”

陈明说：“我的苹果数不是最少的，李小明和我的苹果数差 3 个，李小明有  $b$  个苹果。”

李小明说：“我比刘红苹果少，刘红有  $c$  个苹果，陈明比刘红多 3 个苹果。”

他们每人说的三句话中，都有一句是错话.请问：他们各有多少苹果

输入  $a b c$

输出 他们各有多少苹果

样例输入 22 25 23

样例输出 22 25 21

写出表达式

刘红说：“我有  $a$  个苹果，比陈明少 2 个，比李小明多 1 个。”  
\_\_\_\_\_

陈明说：“我的苹果数不是最少的，李小明和我的苹果数差 3 个，李小明有  $b$  个苹果。”  
\_\_\_\_\_

李小明说：“我比刘红苹果少，刘红有  $c$  个苹果，陈明比刘红多 3 个苹果。”  
\_\_\_\_\_

他们每人说的三句话中，都有一句是错话  
\_\_\_\_\_

#### 5 [7741]地理课\*

地理课上老师挂出一张没有注明省份的中国地图.其中有 5 个省份分别编上了数字 1~5 号，  
请同学们写出每个编号是哪一省。（用数组实现比较好）

A 答：2 号是陕西，5 号是甘肃；

B 答：2 号是湖北，4 号是山东；

C 答：1 号是山东，5 号是吉林；

D 答：3 号是湖北，4 号是吉林；

E 答：2 号是甘肃，3 号是陕西。

这 5 名同学每人都只答对了一个省，并且每个编号只有一个人答对。问从 1 号到 5 号各是哪个省

分析：先假设 A 说的第 1 句话正确，即“2 号是陕西”，由 E 的回答知“2 号是甘肃”不正确，从而只能“3 号是陕西”正确，这与假设矛盾！所以只能推定 A 说的第二句话“5 号是甘肃”正确，下面只需利用条件“5 名同学只答对了一个省及每个编号只由 1 个人答对”，推出正确的结果 1 号是山东,2 号是湖北,3 号是陕西,4 号是吉林,5 号是甘肃。

写出表达式：

A 答：2 号是陕西，5 号是甘肃； \_\_\_\_\_

B 答：2 号是湖北，4 号是山东； \_\_\_\_\_

C 答：1 号是山东，5 号是吉林； \_\_\_\_\_

D 答：3 号是湖北，4 号是吉林； \_\_\_\_\_

E 答：2 号是甘肃，3 号是陕西。 \_\_\_\_\_

这 5 名同学每人都只答对了一个省，并且每个编号只有一个人答对

\_\_\_\_\_

## 习题

1.

```
#include <iostream>
using namespace std;
int main ()
{
    int a=10,b=29,c=5,d,e;
    d=(a+b)/c;
    e=(a+b)%c;
    cout<<"d="<<d<<","e="<<e<<endl;    结果是： _____
}

```

2.

```
#include<iostream>
using namespace std;
int a,b,c,ans = 100;
int main(){
    a = 1; b = 2; c = 5;
    a = a + b;
    b = a + b + c;
    ans = ans + a + b + c;
    cout << ans << endl;
}

```

```
return 0;
}
```

结果是：\_\_\_\_\_

3. 歌手大奖赛上 6 名评委给一位参赛者打分，6 个人打分的平均分为 9.6 分；如果去掉一个最高分，这名参赛者的平均分为 9.4 分；如果去掉一个最低分，这名参赛者的平均分为 9.8 分；如果去掉一个最高分和一个最低分，这名参赛者的平均是多少？

```
#include<cstdio>
int main()
{
    double high,low,sc_all,sc_high,sc_low,ans;
    _____1_____ //求 6 名评委的总分
    sc_high=5*9.4; //求去掉最高分后的总分
    _____2_____ //求去掉最低分后的总分
    high=sc_all-sc_high; //求最高分
    low=_____3_____ //求最低分
    ans=_____4_____ //求平均分
    printf("%.2f\n",ans); //%.2f 按实数格式输出，保留 2 位小数
}
}
```

4. 有以下程序：

```
main( )
{
    int a=2,b=5,c=3,d=2,x;
    if(a%3>b)
        if(c>d)
            if(b<d) x=++b;
            else x=++d;
        else x=--c;
    else x=++b;
}
}
```

程序运行后 x 的值是( )

A)6            B)3            C)2            D)7

4. 有以下程序：

```
main()
{
    int x,y;
    cin>>x;
    if(x>5)
        if(x<8)
            y=x*x-1;
        else if(x!=10)
            y=3*x+1;
        else
            y=10;
    else
        y=x;
    printf("%d\n",y);
}
```

程序运行后输入 11<回车>, 程序的输出结果是( )

A)120

B)34

C)10

D)11

6.

```
#include <iostream>
using namespace std;
int main()
{
    int x, y, z, a;
    x=123;
    y=232;
    z=x%10;
    a=y%15;
    x=z+a;
    z=x-z, a=x-z;
    cout<<z<<" "<<a;
    return 0;
}
```

输出: \_\_\_\_\_

7.

```
#include <iostream>
using namespace std;
int main()
{
    int x, a, b, c, d, z;
    cin >> x;
    a = x % 10;
    b = (x % 100) / 10;
    c = (x % 1000) / 100;
    d = x / 1000;
    z = a * a + b * b + c * c + d * d;
    if ((c * d) % (a * b) == 0)
        z += a * b;
    else
        z -= c * d;
    cout << z << endl;
    return 0;
}
```

输入:

1234

输出: \_\_\_\_\_

8.

```
#include <iostream>
using namespace std;
int main()
{
    int a,b,c;
    a=45;
    b=23;
    if(a>b)
        cout<<"a>b";
    else
        cout<<"a<=b";
    return 0;
}
```

输出: \_\_\_\_\_

9.

```
#include <iostream>
using namespace std;
int main()
{
    int x, y, t;
    cin >> x >> y;
    if (x < y)
    {
        t = x;
        x = y;
        y = t;
    }
    cout << x << " " << y;
    return 0;
}
```

输入: 34 52

输出: \_\_\_\_\_



# 第九讲 循环初探

本讲内容:

- 1. 理解循环结构的含义
- 2. 掌握 while 循环
- 3. 掌握理解条件表达式

## 1.1 知识剖析

如果充当条件判断的表达式为真(非 0)，执行循环体语句；循环体执行完后又返回到循环条件判断处；如果循环条件仍然成立，重复上述过程；如果循环条件不成立，则整个循环结构执行完毕。

循环结构如下：

<pre>while(表达式)     语句 1;</pre>	<pre>while(表达式){     语句 1;     语句 2; }</pre>	
---------------------------------	--	--

其含义为：先计算表达式（一般称为循环条件）的值，当表达式的值为真（循环条件成立）时，去执行一次循环体。和 if 语句不同的是，执行完一次循环体后，while 语句又回到开始处，继续计算和判断表达式的真假，决定是否再次执行循环体。也就是“当表达式成立时，不断重复执行循环体”。

填空：理解 while 循环后,写出 1,2,3,4 的含义

```
1// _____
while(2){// _____
    3// _____
    5// _____
}
```

随堂练习 1

- 1. while 语句中循环结束的条件是 while 后面表达式的值是\_\_\_\_\_。  
A)0                      B)1                      C)-1                      D)非 0
- 2. int k=1;  
while(k<=10)k++; 该程序段结束后, k 的值为(            )  
A)10                      B)9                      C)8                      D)11
- 3.有以下程序段  
int x=1;  
while(x==0){  
    x=x\*x;

- } 则该程序段\_\_\_\_\_。
- A)是死循环    B)循环执行 1 次    C) 循环执行 2 次    D)循环执行 0 次

## 1.2 案例分析

### 1[7124] 计算好天气

如果认为温度大于 20 度就是好天气，输入周一到周日的每天的温度，统计好天气的天数。输入 7 天的温度输出好天气的天数

样例输入 21 18 19 23 22 16 23

样例输出 4

分析：

1. 如何表示温度大于 20 度就是好天气？

定义： 一个变量 t 表示温度；  
一个变量 sum 表示天数  
sum 初始化为 0，即 sum=0;  
if(t>20)sum++;

2. 如何实现连续输入每天的温度？

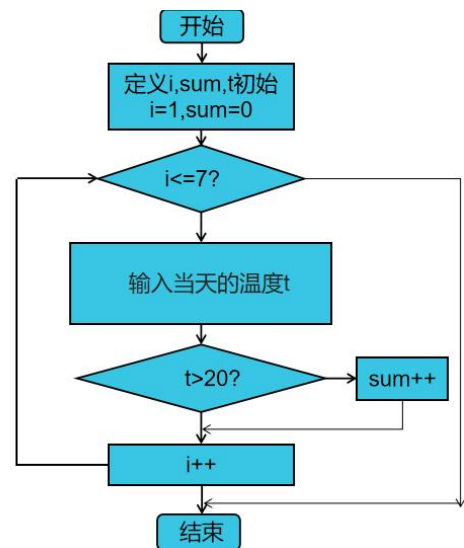
```
while(条件成立){
    cin>>t;
    i++;
}
```

2. 如何实现判断一周 7 天的温度？

```
i=1;
while(i<=7){
    .....
    i++;
}
```

完成代码：

```
#include<iostream>
using namespace std;
int main()
{
    int i=1,sum=0,t;//1 初始化
    while(i<=7){//2 循环条件
        cin>>t;//循环语句
        if(t>20)sum++;
    }
```



```

        i++; //更新循环变量
    }
    cout<<sum;
    return 0;
}

```

## 2[2195]求和

输入一个正整数  $n$ ，计算  $1+2+\dots+n$

分析：如何求和？如何实现  $n$  次相加？

1. 初始状态是什么\_\_\_\_\_
2. 循环的结束条件时什么\_\_\_\_\_
3. 每次循环做什么事情\_\_\_\_\_

流程图：

```

#include<iostream>
using namespace std;
int main(){
    int n,i,sum;//步定义变量
    cin>>n;
    i=1,sum=0; // 1 初始化
    while(i<=n){ //2 确定循环条件
        sum=sum+i; //3 计算
        i++; //4 修改循环变量
    }
    cout<<sum;
}

```

### 随堂练习 2

1. 有以下程序：

```

main( )
{
    int a=5,sum=0;
    while(a>=1)
    { sum=sum+a; a--; }
    printf("%d",sum); //cout<<sum;
}

```

程序运行后的输出结果是\_\_\_\_\_。

- A)15                      B)14                      C)5                      D)0

2 [1019] 编程实现：累加 100~1000 之内能被 3 整除的偶数。

```

#include <iostream>
using namespace std;
int main()
{
    int i,sum;
    ___1_____// 1 初始化
    while(___2_____){//2 确定循环条件
        if(i%6==0)
            ___3_____//3 计算
            ___4_____ //4 修改循环变量
    }
    cout<<sum;
    return 0;
}

```

### 3 阅读程序

写出程序的运行结果\_\_\_\_\_

```

#include<iostream>
using namespace std;
int main(){
    int i = 100,x = 0,y = 0;
    while(i > 0){
        i--;
        x = i % 8;
        if(x == 1) y++;
    }
    cout << y << endl;
    return 0;
}

```

**问题分析:** 一般采用列表观察、找规律的方法，阅读和理解程序。对于本题而言，循环条件为“i > 0”，用到 i、x 和 y 这 3 个变量，列表如表。

i 的取值	循环条件	循环体执行情况
100	非 0	i 更新为 99,x 赋值为 3,y 的值不变还是 0
99	非 0	i 更新为 98,x 赋值为 2,y 的值不变还是 0
98	非 0	i 更新为 97,x 赋值为 1,y 的值变为 1
97	非 0	i 更新为 96,x 赋值为 0,y 的值不变还是 1
96	非 0	i 更新为 95,x 赋值为 7,y 的值不变还是 1
⋮		
1	非 0	i 更新为 0,x 赋值为 0,y 的值不变还是 13
0	0	退出循环

通过找规律发现，每次 i 是  $8 \times x + 1$  时，y 的值会加 1，也就是 97、89、81、...、9、1，一共  $99/8 + 1 = 13$ 。所以，程序最后输出 y 的值为 13。

## 课前练习

<p>1 阅读程序</p> <pre>#include &lt;iostream&gt; using namespace std; int main() {     int x, y, s, p;     cin &gt;&gt; x &gt;&gt; y;     s = x + y;     p = y - x;     s = s - p;     cout &lt;&lt; s &lt;&lt; endl;     return 0; }</pre> <p>输入: 13 31 输出:</p>	<p>2. 阅读程序</p> <pre>#include &lt;iostream&gt; using namespace std; int main() {     int x, y;     cin &gt;&gt; x &gt;&gt; y;     x = x + y;     y = x - y;     x = x - y;     cout &lt;&lt; x &lt;&lt; " " &lt;&lt; y &lt;&lt; endl;     return 0; }</pre> <p>输入: 3 7 输出:</p>
--	---

## 习题

1. while 语句中循环结束的条件是 while 后面表达式的值是( )
- A)0                      B)1                      C)-1                      D)非 0
2. 程序段如下：则以下说法中正确的是：\_\_\_\_\_。

```
int k=-20;
while(k=0) k=k+1;
```

- A. while 循环执行 20 次                      B. 循环是无限循环  
C. 循环体语句一次也不执行                      D. 循环体语句执行一次

3. 有以下程序:

```
int main()
{
    int number=0;
    while(number<= 1){
        cout<<"*"<<number; //printf("%d", number);
        number++;
    }
    cout<<"**"<<number<<endl;//printf("**%d\n", number);
    return 0;
}
```

程序运行后的输出结果是\_\_\_\_\_ D

- A)\*1\*\*2                      B)\*1\*2\*\*3                      C)\*1\*2                      D)\*0\*1\*\*2

<p>4. 以下程序执行后的输出结果是_____。</p> <pre>main( ) {     int s=10;     while(s--); s+=3;     printf("%d",s); }</pre>	<p>5. 以下程序执行后的输出结果是_____</p> <pre>main() {     int k=0;     while(k==0)         k--;     printf("k=%d",k); }</pre>
--	--

6. 以下程序执行后的输出结果是\_\_\_\_\_

```
main()
{
    int n=12345,s=0;
    while(n)
    {
        s++;
        n=n/10;
    }
    printf("%d",s);
}
```

7. 以下程序的功能是：从键盘输入的若干整数中，分别统计正整数和负整数的个数，用 0 结束输入操作。请填空。

```
main()
{
    int a, _____ 1 _____;
    scanf("%d",&a);
    while(_____ 2 _____)
    {
        if (a>0)    n1++;
        if (a<0)    n2++;
        _____ 3 _____;
    }
    printf("%d,%d\n",n1,n2);
}
```

# 第十讲 表达式综合

## 1 [7743] 阶梯电价

为了提倡居民节约用电，某省电力公司执行“阶梯电价”，安装一户一表的居民用户电价分为两个“阶梯”：月用电量 50 千瓦时（含 50 千瓦时）以内的，电价为 0.53 元/千瓦时；超过 50 千瓦时的，超出部分的用电量，电价上调 0.05 元/千瓦时。请编写程序计算电费。

输入格式：

输入在一行中给出某用户的月用电量（单位：千瓦时）。

输出格式：

在一行中输出该用户应支付的电费（元），结果保留两位小数，格式如：“cost = 应付电费值”；若用电量小于 0，则输出“Invalid Value!”。

输入样例 1:

10

输出样例 1:

cost = 5.30

输入样例 2:

100

输出样例 2:

cost = 55.50

输入

输出

样例输入

10

样例输出

cost = 5.30

```
#include<iostream>
using namespace std;
int main()
{
    int x;
    double y;
    cin>>x;
    if(x<0)
        cout<<"Invalid Value!";
    else if(x<50) {
        y=0.53*x;//输出注意格式，最好是直接从题目拷贝
        printf("cost = %.2lf",y);
    }
    else {
        y=26.5+(x-50)*0.58;
        printf("cost = %.2lf",y);
    }
}
```

```
    }  
    return 0;  
}
```

## 2 [2164] 计算个人所得税

输入月薪 salary，输出应交的个人所得税 tax（保留 2 位小数）。按照 2011 年开始的新的个人所得税法，计算公式为：

$$\text{tax}=\text{rate}*(\text{salary}-3500)-\text{deduction}$$

当 salary≤3500 时，rate=0、deduction=0

当 3500<salary≤5000 时，rate=3%、deduction=0

当 5000<salary≤8000 时，rate=10%、deduction=105

当 8000<salary≤12500 时，rate=20%、deduction=555

当 12500<salary≤38500 时，rate=25%、deduction=1005

当 38500<salary≤58500 时，rate=30%、deduction=2755

当 58500<salary≤83500 时，rate=35%、deduction=5505

当 83500<salary 时，rate=45%、deduction=13505

输入

月薪 salary

输出

个人所得税 tax

样例输入

4238.9

7328.6

52547

89000

样例输出

22.17

277.86

11959.10

24970.00

```
#include<bits/stdc++.h>  
using namespace std;  
int main(){  
    double tax,deduction,rate,salary;  
    while(cin>>salary){  
        if(salary<0)break;  
        if(salary<=3500&&salary>=0)  
            rate=0,deduction=0;  
        else if(salary>3500&&salary<=5000)
```



```

        rate=0.03,deduction=0;
    else if(salary>5000&&salary<=8000)
        rate=0.1,deduction=105;
    else if(salary>8000&&salary<=12500)
        rate=0.2,deduction=555;
    else if(salary>12500&&salary<=38500)
        rate=0.25,deduction=1005;
    else if(salary>38500&&salary<=58500)
        rate=0.3,deduction=2755;
    else if(salary>58500&&salary<=83500)
        rate=0.35,deduction=5505;
    else if(salary>83500)
        rate=0.45,deduction=13505;

    tax=rate*(salary-3500)-deduction;
    cout.precision(2);
    cout<<fixed<<tax<<endl;
}
return 0;
}

```

### 3[3228] 小 Q 买笔

期末来临了，班长小 Q 决定将剩余班费  $x$  元钱，用于购买若干支钢笔奖励给一些学习好、表现好的同学。已知商店里有三种钢笔，它们的单价为 6 元、5 元和 4 元。小 Q 想买尽量多的笔（鼓励尽量多的同学），同时他又不想有剩余钱。请您编一程序，帮小 Q 制订出一种买笔的方案。

输入

输入班费的数量

输出

输出 6 元、5 元和 4 元钱笔的数目

样例输入

27

样例输出

1 1 4

```

#include<iostream>
using namespace std;
int main()
{
    int a,b,c,x,y;
    cin>>x;
    c=x/4;
    y=x%4;
}

```

```

switch(y)
{
    case 0:a=0;b=0;break;
    case 1:a=0;b=1;c--;break;
    case 2:a=1;b=0;c--;break;
    case 3:a=1;b=1;c-=2;break;
}
cout<<a<<' '<<b<<' '<<c<<endl;
return 0;
}

```

## 4 [2613] 点和正方形的关系

题目描述

有一个正方形，四个角的坐标  $(x,y)$  分别是  $(1, -1)$ ,  $(1, 1)$ ,  $(-1, -1)$ ,  $(-1, 1)$ ,  $x$  是横轴,  $y$  是纵轴。写一个程序，判断一个给定的点是否在这个正方形内(包括正方形边界)。如果点在正方形内，则输出 **yes**，否则输出 **no**。

输入

输入一行，包括两个整数  $x$ 、 $y$ ，以一个空格分开，表示坐标  $(x,y)$ 。

输出

输出一行，如果点在正方形内，则输出 **yes**，否则输出 **no**。

样例输入

1 1

样例输出

yes

```

#include<iostream>
using namespace std;
int main()
{
    int x,y;
    cin>>x>>y;
    if(x<=1&&x>=-1&&y>=-1&&y<=1)
        cout<<"yes"<<endl;
    else
        cout<<"no"<<endl;
    return 0;
}

```

# 1 while 循环

本讲内容:

4. 理解循环结构的含义
5. 掌握 while 循环
6. 掌握理解条件表达式

## 1.1 知识剖析

如果充当条件判断的表达式为真(非 0)，执行循环体语句；循环体执行完后又返回到循环条件判断处；如果循环条件仍然成立，重复上述过程；如果循环条件不成立，则整个循环结构执行完毕。

循环结构如下：

<pre>while(表达式)     语句 1;</pre>	<pre>while(表达式){     语句 1;     语句 2; }</pre>	
---------------------------------	--	--

其含义为：先计算表达式（一般称为循环条件）的值，当表达式的值为真（循环条件成立）时，去执行一次循环体。和 if 语句不同的是，执行完一次循环体后，while 语句又回到开始处，继续计算和判断表达式的真假，决定是否再次执行循环体。也就是“当表达式成立时，不断重复执行循环体”。

填空：理解 while 循环后,写出 1,2,3,4 的含义

```
1//_____
while(2){//_____
    3//_____
    5//_____
}
```

随堂练习 1

1. while 语句中循环结束的条件是 while 后面表达式的值是\_\_\_\_\_。

- A)0                      B)1                      C)-1                      D)非 0

2. int k=1;

while(k<=10)k++; 该程序段结束后, k 的值为( )

- A)10                      B)9                      C)8                      D)11

3.有以下程序段

```
int x=1;
while(x==0){
```

- `x=x*x;`  
} 则该程序段\_\_\_\_\_。  
A)是死循环    B)循环执行 1 次    C) 循环执行 2 次    D)循环执行 0 次

## 1.2 案例分析

### 1[7124] 计算好天气

如果认为温度大于 20 度就是好天气，输入周一到周日的每天的温度，统计好天气的天数。输入 7 天的温度输出好天气的天数  
 样例输入 21 18 19 23 22 16 23  
 样例输出 4

分析：

3. 如何表示温度大于 20 度就是好天气？

定义： 一个变量 t 表示温度；  
一个变量 sum 表示天数  
sum 初始化为 0，即 sum=0;  
if(t>20)sum++;

2. 如何实现连续输入每天的温度？

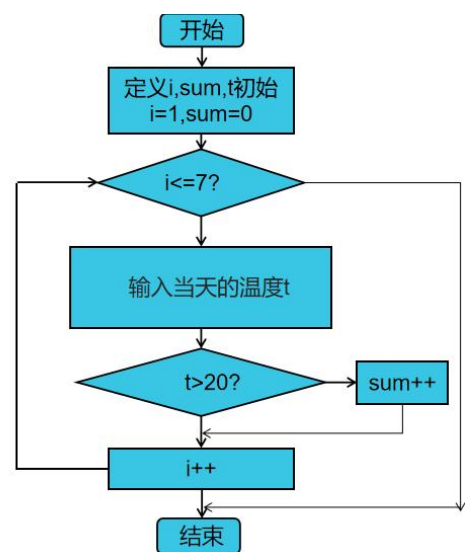
```
while(条件成立){
    cin>>t;
    i++;
}
```

4. 如何实现判断一周 7 天的温度？

```
i=1;
while(i<=7){
    .....
    i++;
}
```

完成代码：

```
#include<iostream>
using namespace std;
int main()
{
    int i=1,sum=0,t;//1 初始化
    while(i<=7){//2 循环条件
        cin>>t;//循环语句
```



```

        if(t>20)sum++;
        i++;//更新循环变量
    }
    cout<<sum;
    return 0;
}

```

## 2[2195]求和

输入一个正整数 n，计算 1+2+.....+n

分析：如何求和？如何实现 n 次相加？

4. 初始状态是什么\_\_\_\_\_
5. 循环的结束条件是什么\_\_\_\_\_
6. 每次循环做什么事情\_\_\_\_\_

```

#include<iostream>
using namespace std;
int main(){
    int n,i,sum;//步定义变量
    cin>>n;
    i=1,sum=0; // 1 初始化
    while(i<=n){//2 确定循环条件
        sum=sum+i;//3 计算
        i++; //4 修改循环变量
    }
    cout<<sum;
}

```

流程图：

### 随堂练习 2

1. 有以下程序：

```

main( )
{
    int a=5,sum=0;
    while(a>=1)
    { sum=sum+a; a--; }
    printf("%d",sum);//cout<<sum;
}

```

程序运行后的输出结果是\_\_\_\_\_。

A)15

B)14

C)5

D)0

**2 [1019]** 编程实现：累加 100~1000 之内能被 3 整除的偶数。

```
#include <iostream>
using namespace std;
int main()
{
    int i,sum;
    1 // 1 初始化
    while(2){ // 2 确定循环条件
        if(i%6==0)
            3 // 3 计算
            4 // 4 修改循环变量
    }
    cout<<sum;
    return 0;
}
```

**3 阅读程序**

写出程序的运行结果\_\_\_\_\_

```
#include<iostream>
using namespace std;
int main(){
    int i = 100,x = 0,y = 0;
    while(i > 0){
        i--;
        x = i % 8;
        if(x == 1) y++;
    }
    cout << y << endl;
    return 0;
}
```

**问题分析：**一般采用列表观察、找规律的方法，阅读和理解程序。对于本题而言，循环条件为“i > 0”，用到 i、x 和 y 这 3 个变量，列表如表。

i 的取值	循环条件	循环体执行情况
100	非 0	i 更新为 99,x 赋值为 3,y 的值不变还是 0
99	非 0	i 更新为 98,x 赋值为 2,y 的值不变还是 0
98	非 0	i 更新为 97,x 赋值为 1,y 的值变为 1
97	非 0	i 更新为 96,x 赋值为 0,y 的值不变还是 1
96	非 0	i 更新为 95,x 赋值为 7,y 的值不变还是 1
⋮		
1	非 0	i 更新为 0,x 赋值为 0,y 的值不变还是 13
0	0	退出循环

通过找规律发现，每次 i 是  $8*x+1$  时，y 的值会加 1，也就是 97、89、81、...、9、1，一共  $99/8+1=13$ 。所以，程序最后输出 y 的值为 13。

### 3 [2206] 求各位数字之和

输入一个整数，求它的各位数字之和。

样例输入 -234

样例输出 9

分析：

求一个正整数的各位数字之和，需要“分解”出它的每一位数字，每次进行累加。定义累加器  $s$ ，初始化为 0。对于正整数  $n$ ，分解的过程就是从低位到高位，不断求余累加 ( $s += n \% 10$ ) 和整除 ( $n = n / 10$ )，直到  $n$  等于 0。

1. 初始状态是什么\_\_\_\_\_
2. 循环的结束条件是什么\_\_\_\_\_
3. 每次循环做什么事情\_\_\_\_\_

```
#include<iostream>
using namespace std;
int main(){
int sum,n;
cin>>n;
if(n<0)n=-n;
while(_1_____){
    sum=sum+_2_____
    _3_____}
cout<<sum;
return 0;
}
```

#### 随堂练习 3

1.有以下程序：

```
main( )
{   int  x=100,y=10,z=3;
    while(x--)
    {   y+=x--; z=z+x/3;   }
    printf("%d",x);
}
```

程序运行后的输出结果是\_\_\_\_\_。

- A)0                      B)1                      C)-1                      D)100

2. 以下程序执行后的输出结果是\_\_\_\_\_。

```
main()
{   int num=0;
    while(num<=2)
    {   num++;
```

```
        printf("%d ",num);
    }
}
```

### 3. 求平均值

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    double a,b,x;
    double sum=0,ave; //求和初始值
    int i,n;
```



```
ave=sum/n;
cout.precision(4);
cout<<fixed<<ave;
return 0;
}
```

## 4 [1048] 统计输出人数

编写程序，输入一批学生的成绩，遇 0 或负数则输入结束，要求统计并输出优秀（大于等于 85）、通过（60~84）和不及格（小于 60）的学生人数。

样例输入

88 71 68 70 59 81 91 42 66 77 83 0

样例输出

>=85:2

60-84:7

<60:2

思考：

1. 循环结束的条件是什么？ \_\_\_\_\_
2. 如何实现输入多个数？ \_\_\_\_\_



```

#include<iostream>
using namespace std;
int main(){
    int i,x1=0,x2=0,x3=0,n;
    1 _____ // cin>>n;
    while(n>0){//n>0
        if(2 _____)x1++;
        else if(n>=60)x2++;
        else
            x3++;
        3 _____
    }
    cout<<">=85:"<<x1<<endl;
    cout<<"60-84:"<<x2<<endl;
    cout<<"<60:"<<x3<<endl;
    return 0;
}

```

## 6 [7708] 停车收费

### 题目描述

随着时间的进步，我们的生活越来越智能化，越来越多的人工智能运用到了我们生活中，像以前还需要人来收取停车费，现在已经实现了无人化收费，现在在王充路上的停车场收费如下

- 停车时间在 0.5 小时及以内，则收费 0 元
- 停车时间超过 0.5 小时但在 1 小时及以内，收费 2 元
- 超过 1 小时部分，每小时收费 3 元（不足 1 小时按 1 小时收取）

停车场可以自动记录每辆车进入时间和出去时间。我们要根据记录仪里面停车的进入时间和出去时间计算停车场的一辆车停车时间（有效时间，在收费时段中的时间）。计算这辆车停车的收费。

输入：n(n<100)辆车**进入时间和出去时间**，时间是多少小时多少分钟，即四个数据，a,b,c,d，表示 a 小时 b 分钟进入，c 小时 d 分钟离开 (0<=小时<=24) (0<=分钟<=60)

输出：一行一个整数，停车价格

样例输入

```

4
19 56 21 57
17 56 20 12

```



20 00 23 00

8 56 9 56

样例输出

2

8

0

2

分析：

1. 如何输入 n 辆车？	2. 如何计算每辆车停的时间？	3 需要分哪几种情况讨论收费？
---------------	-----------------	-----------------

```
#include <iostream>
using namespace std;
int main(){
    int n,i=1;
    int h1,h2,m1,m2,money,M1,M2,H;
    double M,H1;
    cin>>n;
    while(i<=n){
        cin>>h1>>m1>>h2>>m2;//19 56~21 57
        M1=_____1____; //进入的时间： 分钟
        M2=_____2____; //出去的时间： 分钟
        if(M1<480){
            M1=480;//8 点前
        }
        if(M2>1230){
            M2=1230;//20:30 以后
        }
        M=_____3____ //停车时间
        if(_____4_){//不到半小时
            money=0;
        }
        else if(_____5_){//不到 1 小时
            money=2;
        }
        else{
            H1=M/60;//H1 是 double
        }
    }
}
```

```

        H=H1;
        if(H1>H){//如果不能整除
            H++;
        }
        money=_____6__;
    }
    cout<<money<<endl;
    i++;
}
return 0;
}

```

### 1.3 思维拓展：等差数列

若干个数排成一列称为数列。数列中的每一个数称为一项。其中第一项称为首项，最后一项称为末项，数列中项的个数称为项数。

从第二项开始，后项与其相邻的前项之差都相等的数列称为等差数列，后项与前项的差称为公差。

[例 1] 有一个数列：4，10，16，22，…，52.这个数列共有多少项？

思路：容易看出这是一个等差数列，公差为 6，首项是 4，末项是 52.要求项数，可直接带入项数公式进行计算。

项数 =  $(52 - 4) \div 6 + 1 = 9$ ，即这个数列共有 9 项。

项数 =  $(\text{末项} - \text{首项}) \div \text{公差} + 1 = (52 - 4) \div 6 + 1 = 48 \div 6 + 1 = 9$

#### 7 [1072] 等差数列

给定一个序列 2 5 8 11 14...，输入正整数 n，求前 n 项和。

样例输入 2

样例输出 7

思考：

1 初始状态是什么\_\_\_\_\_

2 循环的结束条件时什么\_\_\_\_\_

3 每次循环做什么事情\_\_\_\_\_

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n,i,item=2,sum=0;
```

```
    cin>>n;
```

```
    _____//1 初始化
```

```
    while(i<=n){
```

```
        sum=_____;//2
```

```

        _____://3
    }
    cout<<sum;
    return 0;
}

```

### 课堂作业列表

- |                   |                 |
|-------------------|-----------------|
| [7124] 计算好天气      | [2195]求和        |
| [2206] 求各位数字之和    | [7708] 停车收费     |
| [1019] 累加偶数       | [2207] 求一个整数的位数 |
| [2208] 各位数字之和     | [2649] 数字反转     |
| [2650] 含 k 个 3 的数 | [3323: 小玉在游泳    |

### 课前练习

<p>1 阅读程序</p> <pre> #include &lt;iostream&gt; using namespace std; int main() {     int x, y, s, p;     cin &gt;&gt; x &gt;&gt; y;     s = x + y;     p = y - x;     s = s - p;     cout &lt;&lt; s &lt;&lt; endl;     return 0; } </pre> <p>输入: 13 31</p> <p>输出:</p>	<p>2. 阅读程序</p> <pre> #include &lt;iostream&gt; using namespace std; int main() {     int x, y;     cin &gt;&gt; x &gt;&gt; y;     x = x + y;     y = x - y;     x = x - y;     cout &lt;&lt; x &lt;&lt; " " &lt;&lt; y &lt;&lt; endl;     return 0; } </pre> <p>输入: 3 7</p> <p>输出:</p>
---	--

### 习题

1. while 语句中循环结束的条件是 while 后面表达式的值是( )
- A)0                      B)1                      C)-1                      D)非 0

6. 程序段如下：则以下说法中正确的是：\_\_\_\_\_。

```

int k=-20;
while(k=0) k=k+1;

```

- |                    |              |
|--------------------|--------------|
| A. while 循环执行 20 次 | B. 循环是无限循环   |
| C. 循环体语句一次也不执行     | D. 循环体语句执行一次 |

7. 有以下程序:

```

int main()
{
    int number=0;
    while(number<= 1){
        cout<<"*"<<number; //printf("%d", number);
        number++;
    }
    cout<<"**"<<number<<endl;//printf("**%d\n", number);
    return 0;
}

```

程序运行后的输出结果是\_\_\_\_\_ D

A)\*1\*\*2                      B)\*1\*2\*\*3                      C)\*1\*2                      D)\*0\*1\*\*2

<p>8. 以下程序执行后的输出结果是_____。</p> <pre> main( ) {     int s=10;     while(s--); s+=3;     printf("%d",s); } </pre>	<p>9. 以下程序执行后的输出结果是_____</p> <pre> main() {     int k=0;     while(k==0)         k--;     printf("k=%d",k); } </pre>
--	--

6. 以下程序执行后的输出结果是\_\_\_\_\_

```

main()
{
    int n=12345,s=0;
    while(n)
    {
        s++;
        n=n/10;
    }
    printf("%d",s);
}

```

7. 以下程序的功能是：从键盘输入的若干整数中，分别统计正整数和负整数的个数，用 0 结束输入操作。请填空。

```

main()
{
    int a,_____ 1 _____;
    scanf("%d",&a);
    while(_____ 2 _____)
    {
        if (a>0)    n1++;
        if (a<0)    n2++;
        _____ 3 _____;
    }
    printf("%d,%d\n",n1,n2);}

```

## 2 等差序列

本讲内容:

1. 理解等差数列的含义
2. 掌握用循环求解等差数列的每一项
3. 掌握用循环求解等差数列的和

### 1.1 知识剖析

若个数排成一列称为数列。数列中的每一个数称为一项。其中第一项称为首项，最后一项称为末项，数列中项的个数称为**项数**。从第二项开始，后项与其相邻的前项之差都相等的数列称为**等差数列**，后项与前项的差称为**公差**。

求总和公式：总和=（首项+末项）×项数÷2

第几项公式：第几项=首项+（项数-1）×公差

项数公式：项数=（末项-首项）÷公差+1

[例 1] 有一个数列：4, 10, 16, 22, ..., 52.这个数列共有多少项？

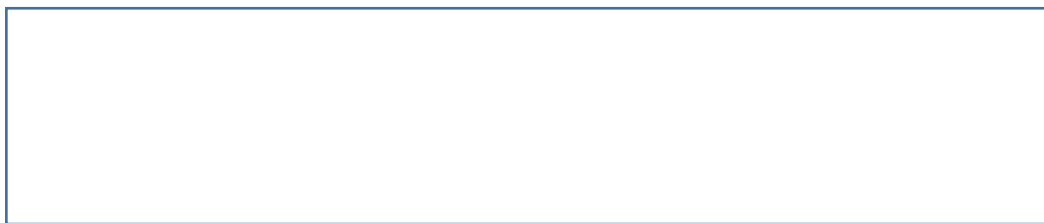
项数=（末项-首项）÷公差+1

=（52-4）÷6+1=48÷6+1=9

思路：容易看出这是一个等差数列，公差为 6，首项是 4，末项是 52.要求项数，可直接带入项数公式进行计算。

项数=（52-4）÷6+1=9，即这个数列共有 9 项。

**编程分析：定义变量，写出表达式求解，如何用循环实现呢？**



[例题 2]计算（2+4+6+...+100）-（1+3+5+...+99）

思路：容易发现，被减数与减数都是等差数列的和，因此，可以先分别求出它们各自的和，然后相减。

进一步分析还可以发现，这两个数列其实是把 1 ~ 100 这 100 个数分成了奇数与偶数两个等差数列，每个数列都有 50 个项。因此，我们也可以把这两个数列中的每一项分别对应相减，可得到 50 个差，再求出所有差的和。

$$\begin{aligned} & (2+4+6+\dots+100) - (1+3+5+\dots+99) \\ &= (2-1) + (4-3) + (6-5) + \dots + (100-99) \\ &= 1+1+1+\dots+1=50 \end{aligned}$$

## 随堂练习 1

1. 等差数列中， 首项=1.末项=39， 公差=2.这个等差数列共有多少项？
2. 求和  $1+3+5+\dots+197+199=$
3. 一个等差数列， 首项=3.公差=2.项数=10， 它的末项是多少？

## 2.2 案例分析

### 1[2540] 等差数列末项计算

#### 题目描述

给出一个等差数列的前两项  $a_1$ ，  $a_2$ ， 求第  $n$  项是多少。

输入一行， 包含三个整数  $a_1$ ，  $a_2$ ，  $n$ 。  $-100 \leq a_1, a_2 \leq 100$ ，  $0 < n \leq 1000$ 。

输出一个整数， 即第  $n$  项的值。

样例输入 1 4 100

样例输出 298

```
#include<iostream>
using namespace std;
int main()
{
    int a1,a2,g,m,x;
    cin>>a1>>a2>>x;
    g=a2-a1;
    m=a1+(x-1)*g;
    cout<<m<<endl;
    return 0;
}
```

### 2 [7619] 等差数列 2

计算形如  $(2+4+6+\dots+n_2) - (1+3+5+\dots+n_1)$  的值。

输入  $n_1, n_2$  ( $n_1, n_2 \leq 10000$ )

输出： 输出表达式的值。

样例输入： 100 99

样例输出： 50

考虑：

1. 初始状态是什么\_\_\_\_\_
2. 循环的结束条件时什么\_\_\_\_\_
3. 每次循环做什么事情\_\_\_\_\_

```
#include<iostream>
using namespace std;
```

```

int main(){
    int n1,n2,i,j,s1=0,s2=0;
    cin>>n1>>n2;
    for(i=__1_____)//i=2;i<=n2;i=i+2
        s1=s1+i;
    for(j=__2_____)//j=1;j<=n1;j=j+2
        s2=s2+j;
    cout<<_3_____;//s2-s1
    return 0;
}

```

### 3 [7620]计算求和

计算:1+11+21+...+31+...n 的和

输入 n 的值, (保证 n 的值满足序列规律, 且 n<3000)

输出 输出表达式的值

样例输入 2011

样例输出 203212

```

#include<iostream>
using namespace std;
int main(){
    int n,__1_____;
    cin>>n;
    while(i<=n){
        sum=_2_____
        _3_____
    }
    cout<<sum;
    return 0;
}

```

### 4 [7621] 数座位数

某体育馆西侧看台有 n 排座位,后面一排都比前面一排多 2 个座位,最后一排有 a 个座位,体育馆西侧看台共有多少个座位?

输入 n 和 a 的值

输出 体育馆西侧看台共有多少个座位

样例输入 30 132

样例输出 3090

谈谈你的思路: \_\_\_\_\_



```

#include<iostream>
using namespace std;
int main(){
    int n,i=1,sum=0,zws;
    cin>>n>>zws;
    while(i<=n){
        sum=_1_____ ;
        _2_____
        _3_____
    }
    cout<<sum;
    return 0;
}

```

### 随堂练习 2

1. 按一定规律排列的算式:4+2,5+8,6+14,7+20,8+26···,那么第 100 个算式是什么?
2. 有 12 个同学聚会,如果见面时每个人都和其余的人握手 1 次,那么一共握手多少次?

### 5 [7622] 青蛙跳

小青蛙沿着台阶往上跳,每跳一次都比上一次升高 4 厘米.它从离地面 10 厘米处开始跳,这一处称为小青蛙的第一个落脚点,如果它的第 n 个落脚点正好在台阶尽头的亭子内,那么这个亭子距地面多少厘米?

输入 输入 n 的值

输出 亭子距地面多少厘米

样例输入 100

样例输出 406

```

#include<iostream>
using namespace std;
int main(){
    int _1_____,n;
    cin>>n;
    while(_2_____){//第 n 的时候没有跳
        h=_3_____ ;
        i++;
    }
    cout<<h;
    return 0;
}

```

## 2.3 思维拓展：奥运奖牌

奥林匹克宪章规定，奥运会的奖牌和奖状由每一届奥运会的组委会提供，但归国际奥委会所有，并且由国际奥委会向获胜运动员颁发。

奥运会的奖牌颁发有个演变的过程。在 1896 年首届现代奥运会上，只向各项目的前两名颁发奖牌，而且是冠军获得银牌，亚军获得铜牌。当时的奖牌由法国艺术家夏普伦设计，直径为 50 毫米。到了 1904 年第三届奥运会，冠军得到了金质奖牌。

1907 年，国际奥委会决定统一奥运会的奖牌式样，两年后对奖牌式样做出如下规定：奖牌正面应为统一图案，反面由当届奥运会组委会自行组织设计。但直到 1928 年第九届奥运会奖牌正面的图案才正式统一，它采用了意大利艺术家卡西奥利的的设计：正面为古罗马竞技场图案，竞技场旁边是左手执象征胜利的棕榈叶、右手举月桂枝的胜利女神像，竞技场的上方是奥运会的届数、地点和年代字样；背面是运动员形象。国际奥委会同时还对奖牌的规格进行了规定：圆形，直径不小于 60 毫米，厚度不小于 3 毫米；冠军和亚军的奖牌必须是银质的，纯度等级至少在 92.5% 以上；冠军奖牌还必须镀有总重量不小于 6 克的纯金；第三名获得铜质奖牌。

2004 年雅典奥运会对奖牌正面图案重新进行了设计，背景由古罗马竞技场改为希腊潘纳辛纳科竞技场全景，胜利女神由原来的坐姿改为动感更强的站姿，并且插上了翅膀。国际奥委会规定，以后的奥运会奖牌正面都将沿用这一图案

此外，国际奥委会还规定，冬季奥运会的奖牌和奖状应同夏季奥运会有所区别，但是没有统一的标准。每届奥运会，不管是夏季还是冬季，奖牌的设计方案都必须得到国际奥委会的首肯。

对于奖牌和奖状的颁发，根据比赛项目和参加人员性质的不同，国际奥委会也有不同的规定。对个人项目，冠军的奖品为银质镀金奖牌和奖状，亚军为银质奖牌和奖状，第三名为铜质奖牌和奖状。奖牌上应当注明获奖运动员所参加比赛的类别和项目，并拴挂在一条可拆卸的项链或绸带上，以便于运动员佩戴。获得第四、第五、第六、第七、第八名的运动员有奖状但无奖牌。如果出现并列第一、第二或第三，则并列的每位运动员都得到奖牌和奖状。

对团体比赛的运动大项和其他运动大项中的团体比赛小项，冠军队的队员，凡是在奥运会期间至少参加了 1 场比赛的，每人都被授予 1 枚金牌和奖状。获第二名的队的每位队员被授予银质奖牌和奖状，获第三名的队的每位队员被授予铜质奖牌和奖状。这些队的其他队员只发给奖状。获得第四、第五、第六、第七和第八名的队的队员获得奖状。

## 6 [2622] 奥运奖牌计数

2008 年北京奥运会，A 国的运动员参与了  $n$  天的决赛项目 ( $1 \leq n \leq 17$ )。现在要统计一下 A 国所获得的金、银、铜牌数目及总奖牌数。输入第 1 行是 A 国参与决赛项目的天数  $n$ ，其后  $n$  行，每一行是该国某一天获得的金、银、铜牌数目。输出 4 个整数，为 A 国所获得的金、银、铜牌总数及总奖牌数。

输入  $n+1$  行，第 1 行是 A 国参与决赛项目的天数  $n$ ，其后  $n$  行，每一行是该国某一天获得的金、银、铜牌数目，以一个空格分开。

输出 1 行，包括 4 个整数，为 A 国所获得的金、银、铜牌总数及总奖牌数，以一个空格分开。

样例输入

3

1 0 3

3 1 0

0 3 0

样例输出

4 4 3 11

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int n,i=1;
    int jp,yp,tp;
    int jpz=0,ypz=0,tpz=0;
    cin>>n;
    while(i<=n){
        cin>>jp>>yp>>tp;
        jpz=jpz+jp;
        ypz=ypz+yp;
        tpz+=tp;
    }
    cout<<jpz<<" "<<ypz<<" "<<tpz<<" "<<jpz+ypz+tpz;
    return 0;
}
```

### 作业列表

[2450] 等差数列末项计算

[7619] 等差数列 2

[7620] 计算求和

[7621] 数座位数

[7622] 青蛙跳

[2622] 奥运奖牌计数

[1073] 查找整除的数

[2630] 人口增长

[2032] 13 的倍数

[1072] 等差数列

### 课前练习

1. 执行以下语句后，输出结果是\_\_\_\_\_。

```
#include<iostream>
using namespace std;
int main()
{
    int i=2,a=23;
    while(i<=5){
        cout<<a%i<<" ";
        i++;
    }
    return 0;
}
```

2 执行以下语句后，输出结果是\_\_\_\_\_。

```
#include<iostream>
using namespace std;
int main(){
{
    int n=12345,s=0;
    while(n)
    {
        s++;
        n=n/10;
    }
    cout<<s;
}
```

--	--

## 习题

1. 语句"while(x%3)a++;"中的表达式 x%3 等价于(            )  
A)x%3!=0            B)x%3==0            C)x%3==1            D)x%3==2

2. 执行以下语句后， a,b 的值分别是\_\_\_\_\_。

```
#include<iostream>
using namespace std;
int main()
{
    int a=0,b=0;
    while(a<4){
        b--;
        a++,b++;
    }
    return 0;
}
```

- A)3 和 0            B)3 和 1            C)4 和 0            D)4 和 1

- 3.执行以下语句后， 输出结果是\_\_\_\_\_。

```
#include<iostream>
using namespace std;
int main(){
{
    int n=12345,s=0;
    while(n)
    {
        s++;
        n=n/10;
    }
    cout<<s;
}
}
```

- 4.填空输入一个整数， 求它的各位数字之和

```
#include<iostream>
using namespace std;
int main(){
int sum,n;
cin>>n;
if(n<0)n=-n;
while(_1_____){
    sum=sum+_2_____;
    _3_____}
cout<<sum;
```

```
return 0;
}
```

5. 已知等比数列的第 1 项  $a_1$  等于 1，公比为 3，以下程序的功能是输出该数列中小于 1000 的最大项  $a_n$  及其对应的  $n$ 。

```
#include<iostream>
using namespace std;
int main()
{
    int n=1,a=1,q=3;
    while(a<=100)
    {
        a=_____ ;
        n++;
    }
    printf("n=%d,a=%d\n",_____, _____);
    //cout<<"n="<<_____<<" ,a="<<_____<<endl;
}
```

6. 执行以下语句后，输出结果是\_\_\_\_\_。

```
#include<iostream>
using namespace std;
int main()
{
    int i=10,k=0;
    while(i<=100){
        if(i>=60)
            k++;
        i=i+5;
    }
    cout<<k;
}
```

# 3 do..while 循环

本讲内容:

- 1. 理解等 do...while 的含义
- 2. 掌握用 do...while 解决循环问题
- 3. 理解 while 和 do...while 的差别

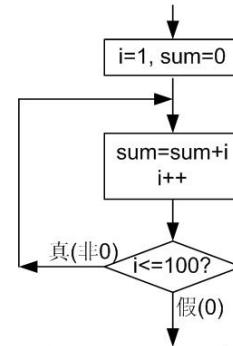
## 3.1 知识剖析

```
do{  
    循环体  
}while (表达式);
```

先执行一次循环体语句，然后判别充当条件判断的表达式；如果表达式为真(表达式的值为非 0)，返回重新执行循环体语句，如此反复，直到表达式的值等于 0 为止，此时循环结束。

假定开学初往帐号上存一定金额，每月花费 800，每月只在月初取出钱，假定不可以透支。问可以取多少次？

- 先查询再取钱 — while 循环；
- 先取钱再查询 — do-while 循环；



### 1[2195]求和

输入一个正整数 n，计算 1+2+.....+n

<pre>#include&lt;iostream&gt; using namespace std; int main(){     int n,i,sum=0;//步定义变量     cin&gt;&gt;n;     i=1,sum=0; // 1 初始化     while(i&lt;=n){//2 确定循环条件         sum=sum+i;//3 计算         i++;      //4 修改循环变量     }     cout&lt;&lt;sum; }</pre>	<pre>#include &lt;iostream&gt; using namespace std; int main( ) {     int i = 1, sum = 0;     do     {         sum = sum+i;         i++;     }while( i&lt;=n );     cout&lt;&lt;sum&lt;&lt;endl; return 0;</pre>
---	--

### 随堂练习 1

1. 有以下程序段

```
x=0;  
do {
```

```
    x++,x=x*x;
} while(!x);
```

则该程序段\_\_\_\_\_。

- A)是死循环    B)循环执行 1 次    C)循环执行 2 次    D)有语法错误

2. 有以下程序段:

```
int a=6;
do
{
    a=a-2;
    printf("%d",a);
} while(--a);
```

则上面程序段\_\_\_\_\_。

- A)输出的是 4    B)输出的是 1    C)输出的是 41    D)是死循环

3. 以下程序运行输出后的结果是\_\_\_\_\_

```
main()
{ int a,num=16118,count=0;
  do
  { a=num%10;
    if(a==1) count++;
    num=num/10;
    while (num);
    printf("count=%d",count);
  }
}
```

## 3.2 案例分析

### 2 [1050]偶数和

编制程序，输入 n 个整数（n 从键盘输入,n>0），输出它们的偶数和。

```
int main()
{ int ____1____,n,m,i=1;
  scanf("%d",&n);
  do
  {
    scanf("%d",&m);
    if(____3____)
      sum=sum+m;
    ____4____
  }
  printf("%d",sum);
}while(____2____);
```

### 3[2647]级数求和

题目描述

已知： $S_n=1+1/2+1/3+\dots+1/n$ 。显然对于任意一个整  $k$ ，当  $n$  足够大的时候， $S_n$  大于  $k$ 。  
现给出一个整数  $k$  ( $1\leq k\leq 15$ )，要求计算出一个最小的  $n$ ，使得  $S_n>k$ 。

输入一个整数  $k$ 。

输出一个整数  $n$ 。

样例输入 1

样例输出 2

**思考：**

1. 初始状态是什么\_\_\_\_\_
2. 循环的结束条件时什么\_\_\_\_\_
3. 每次循环做什么事情\_\_\_\_\_

方法 1:

```
#include<iostream>
using namespace std;
int main(){
    double k,sn=0;
    int i=1;
    cin>>k;
    do{
        sn=sn+1.0/i;
        i++;
    }while(sn<=k);
    cout<<i-1<<endl;
    return 0;
}
```

为什么要减一？

---

方法 2:

```
#include<iostream>
using namespace std;
int main(){
    int k,i=0;
    double s=0;
    cin>>k;
    while(s<=k){
        1
        2
    }
    cout<<i;
```

流程图：



```
    return 0;
}
```

#### 4 [7616] 求 01 数字和

输入一个有 01 组成的十进制整数( $n \leq 1111111111$ 且 $n > 0$ ), 求这个数需要几根火柴棒组成。

输入 一个数

输出 火柴棒的根数

样例输入 10

样例输出 8



思考:

1. 如何取出每位数据?
2. 如何考虑零的情况或者最低位为 0 的情况?
3. 什么情况下用 do...while() 比 while 更方便?

<pre>#include &lt;iostream&gt; using namespace std; int main() {     int a,b,c,x,t,sum=0;     cin&gt;&gt;x; //火柴棍总个数     while(x/10!=0)     // 该数至少有一位     {         t=x%10;         if(t==0)             sum+=6; //加上 0 火柴棍数         else             sum+=2; //加上 1 火柴棍数         x=x/10;     }     if(x==0)         sum+=6;     else</pre>	<pre>#include &lt;iostream&gt; using namespace std; int main() {     int a,b,c,x,t,sum=0;     cin&gt;&gt;x; //火柴棍总个数     do     {         t=x%10;         if(t==0)             sum+=6;         else             sum+=2;         x=x/10;     } while(x!=0);     cout&lt;&lt;sum&lt;&lt;endl;     return 0; }</pre>
--	---

<pre> sum+=2; cout&lt;&lt;sum&lt;&lt;endl; return 0; } </pre>	
---	--

## 5[2646]角谷猜想

谓角谷猜想，是指对于任意一个正整数，如果是奇数，则乘 3 加 1，如果是偶数，则除以 2，得到的结果再按照上述规则重复处理，最终总能够得到 1。如，假定初始整数为 5，计算过程分别为 16、8、4、2、1。程序要求输入一个整数，将经过处理得到 1 的过程输出出来。

输入 一个正整数  $N(N \leq 2,000)$ 。

输出 从输入整数到 1 的步骤，每一步为一行，每一部中描述计算过程。最后一行输出"End"。

如果输入为 1，直接输出"End"。

样例输入

5

样例输出

5\*3+1=16

16/2=8

8/2=4

4/2=2

2/2=1

End

1. 初始状态是什么\_\_\_\_\_
2. 循环的结束条件时什么\_\_\_\_\_
3. 每次循环做什么事情\_\_\_\_\_

```

#include<iostream>
using namespace std;

```

```

int main(){
    int n;//0 变量的定义
    cin>>n;
    //1 初始化
    do{
        if(_2_____){//3-1 循环的内容
            cout<<n<<"*3+1="<<n*3+1<<endl;
            _3_____;//4-1 更新循环变量
        }
        else{
            _4_____;//3-2 循环的内容
        }
    }
}

```

```

        _5 _____; //4-2 更新循环变量
    }
}while(_1 _____); //2 条件
cout<<"End"<<endl;
return 0;
}

```

## 随堂练习 2

1. 以下程序的功能是计算一个整数  $x$  的各位数字之平方和，请选择填空。

```

main()
{ int x,s=0;
  cin>>x;
  do
  { s=s+(x%10)*(x%10);
    x=x/10;
  } while(_____);
  printf("s=%d\n",s);
}

```

- A)  $x < > 0$                       B)  $x == 0$                       C)  $!x$                       D)  $x$

2. 累加 100~1000 之内能被 3 整除的偶数。

```

#include<stdio.h>
int main()
{
  int i=102,s=0;
  do{
    if(i%6==0)
      _____
      _____
  }
  printf("%d",s);
}while( _____1 _____);

```

## 6 [1335]水仙花数

题目描述

输出所有的“水仙花数”。所谓“水仙花数”是指一个 3 位数，其各位数字的立方和等于该数本身。例如，153 是一个“水仙花数”，因为  $153 = 1^3 + 5^3 + 3^3$ 。

从小到大输出所有的“水仙花数”。每行输出一个。

思考：

如何取得一个三位数 NUM 的个位、十位和百位？

```

#include<iostream>
using namespace std;
int main(){
    int a,b,c,n=100;
    do {
        a=_____
        b=_____
        c=_____
        if(n==a*a+a+c*c*c+b*b*b)
            cout<<n<<endl;
        n++;
    }while(n<=999);
    return 0;
}

```

### 3.3 思维拓展：统计

在日常生活中,我们常常需要对一些事物进行统计,例如某一年级中各个班级人数的统计,同一班级中男女生人数的统计,体检中对同学们的身高、体重、视力进行的统计等等。在商场中,为了了解经营情况,商店的经理要对各种商品的销售额逐月进行统计,确定旺季、淡季的供销情况,以获得更好的利润在天气预报中,每天都要对空气中的二氧化硫、二氧化碳、可吸入颗粒物等成分进行搜集、记录并报告有关部门,采取措施,改进空气质量,确保人们的身体健康。

总之,在我们社会生活的各个领域,对各种因素进行调查统计,对于发展生产,提高生活水平是非常重要的.因此,学习一些简单的统计知识,掌握一些初步的统计方法以适应今后的生活及学习是很有必要的。

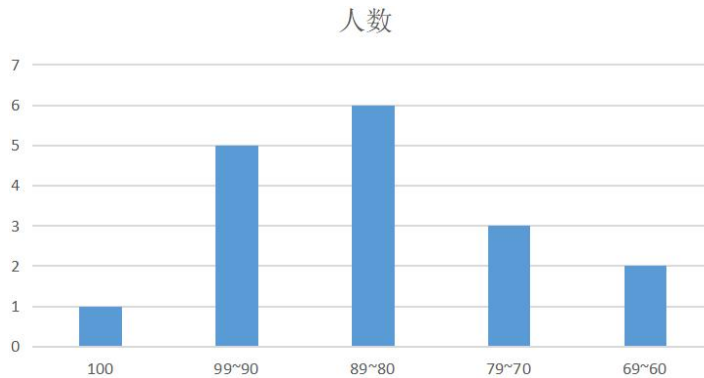
[例 1]为了了解同学们的环保意识,星河小学五年级(1)班环保小分队进行了一次班级环保知识竞赛,竞赛成绩如下:李斌 96 分;张力 83 分;李笑晨 95 分;杨晨 86 分;陈因硕 98 分;周敏 100 分;张力克 84 分;刘薇 67 分;谢静 72 分;罗亦生 81 分;黄亚利 76 分;赵彤杰 88 分;曾浩 65 分;钱可 85 分;张仲华 90 分.这是我们搜集到的数据.请对搜到的数据按分数段进行统计、制成表格,以评估该班级的环保意识状况.

李斌	张力	李笑晨	杨晨	陈因硕	周敏	张力克	刘薇	谢静	罗亦生	黄亚利	赵彤杰	曾浩	钱可	张仲华	沈晓彤	邱宏锐
96	83	95	86	98	100	84	67	72	81	76	88	65	85	90	95	92

统计人数:

分数段	100	99~90	89~80	79~70	69~60
人数	1	5	6	3	2

图像化显示:



### 随堂练习 3

1 某农贸市场各类食品摊位数量统计如表所示:

类别	蔬菜	豆制品	禽蛋	肉	水产品
摊位数 (个)	14	7	8	12	5

- (1) 请将上面统计表制成条形统计图;
- (2) 根据图表说出哪种食品摊位最多? 哪种最少?

### 7 [7617] 分数统计

为了了解同学们的环保意识,星河小学五年级(1)班环保小分队进行了一次班级环保知识竞赛。输入小分队的人数  $n$ , 以及每个人的分数, 依次输出 100,90-99,80-89,70-79,60-69 的人数。(没有不及格的分数)

输入 人数、以及分数

输出 依次输出 100,90-99,80-89,70-79,60-69 的人数

样例输入

17

96 83 95 86 98 100 84 67 72 81 76 88 65 85 76 95 92

样例输出

1 5 6 3 2

```
#include <iostream>
```

```

using namespace std;
int main()
{
    int n,x,i=1,s100=0,s90=0,s80=0,s70=0,s60=0;
    cin>>n;
    do{
        cin>>x;
        if(x==100)s100++;
        else if(x>=90)s90++;
        else if(x>=80)s80++;
        else if(x>70)s70++;
        else s60++;
        i++;
    }while(i<=n);
    cout<<s100<<" "<<s90<<" "<<s80<<" "<<s70<<" "<<s60;
    return 0;
}

```

## 课堂作业列表

[2195] 求累加和    [1050] 偶数和  
 [1335]水仙花数    [7616] 求 01 数字和  
 [2646]角谷猜想    [1335]水仙花数  
 [2175] 求阶乘      [1019] 基础累加  
 [2194] 求累加和    [2647] 级数求和  
 [7617] 分数统计

## 课前练习

1. 读程序写结果\_\_\_\_\_

```

#include <iostream>
using namespace std;
int main()
{   int k=0;
    while(k==0)k--;
    printf("k=%d",k);
    return 0;
}

```

2.执行以下程序段后 m 的值是\_\_\_\_\_

```

#include <iostream>
using namespace std;
int main()
{
    int r,m=9,n=6;

```

```

while (n!=0)
{
    r=m%n;m=n; n=r;
}
cout<<m;
return 0;
}

```

## 习题

1. 有以下程序段:

```

int a=6;
do
{
    a=a-2;
    cout<<a;
} while(--a);

```

则上面程序段\_\_\_\_\_。

A)输出的是 4    B)输出的是 1    C)输出的是 41    D)是死循环

2. 设有以下程序段

```

int a,b=0;
do
{   scanf("%d",&a);
    b=b+a;
}while(a!=100&&b>0);

```

则循环的结束条件是

A)a 的值不等于 100 并且 b 的值大于 0  
 B)a 的值不等于 100 或 b 的值大于 0  
 C)a 的值等于 100 或 b 的值小于等于 0  
 D)a 的值等于 100 并且 b 的值小于等于 0

3. 程序运行后的输出结果是\_\_\_\_\_。

```

#include<iostream>
using namespace std;
int main()
{
    int a=123,b;
    while(a)
    {
        b=a%10;
        cout<<b<<" ";
        a=a/10;
    }
}

```

```
    return 0;
}
```

4. 以下程序运行输出后的结果是\_\_\_\_\_

```
main()
{   int n=12345,sum=0;
    while(n)
    {   sum+=n%10;
        n=n/10;
    }
    printf("%d",sum);
}
```

5. 给定一个长度为 n 的非负整数序列，请计算序列的最大跨度值(最大跨度值 = 最大值减去最小值)。

```
#include<iostream>
using namespace std;
int main()
{   int n,i,x,___1_____;
    scanf("%d",&n);
    while(i<=n){
        scanf("%d",&x);
        if(___2_____)
            max=x;
        if(x<min)
            ___3_____;
        i++;
    }
    printf("%d",___4____);
return 0;
}
```

6. 下面程序的功能是在一个正整数的各位数字中找出最大者。程序中有一个错误，请修改。

```
#include<iostream>
using namespace std;
int main ( )
{
1   int n,max,t;
2   max=9;
3   cin>>n;
4   do
5   {
6       t=n%10;
7       if (max<t)  max=t;
```



```
8     n/=10;
9 }while (!n);
10 cout<<"max="<<max;
11 return 0;
}
```

## 4 for 循环

内容:

1. 理解 for 循环
2. 理解 for 循环各个表达式的用法
3. 学会用 for 语句改写 while 和 do-while 语句。

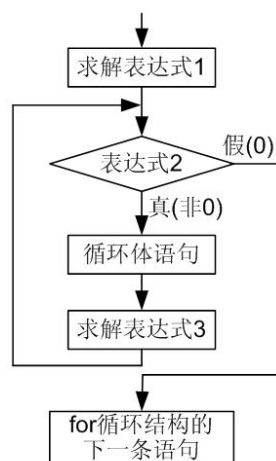
### 4.1 知识剖析

for 语句的格式如下:

```
for( 表达式 1; 表达式 2; 表达式 3){  
    循环语句 4  
}
```

其含义为:

- (1) 先求解表达式 1。
- (2) 再求解表达式 2, 若其值为真(值为非 0), 则执行 for 语句中指定的内嵌的循环语句 4, 然后执行表达式 3。
- (3) 求解表达式 3。
- (4) 转回上面第(2)步骤继续执行若, 为假(值为 0), 则结束循环, 转到第(5)步。为真, 则继续执行内嵌的循环语句 4。
- (5) 循环结束, 执行 for 语句下面的一个语句。



请特别注意 for 循环语句中的 3 个表达式和循环体语句的执行顺序和执行次数。

思考:

1. 初始状态是什么( )
2. 循环的结束条件时什么( )
3. 每次循环做什么事情( )  
A) 表达式 1      B)表达式 2      C) 循环语句 4      D) 表达式 3

for 语句格式举例

- (1)将控制变量从 1 变到 100, 增量为 1  
`for(i=1;i<=100;i++)`
- (2)将控制变量从 100 变到 1, 增量为-1  
`for(i=100;i>=1;i--)`
- (3)控制变量从 7 变到 77, 增量为 7  
`for(i=7;i<=77;__1_____)`
- (4)控制变量从 20 变到 2, 增量为-2  
`for(int i=20;__2_____)`
- (5)按所示数列改变控制变量值: 99、88、77、...、22、11、0, 增量为-11  
`for(int j=99;__3_____)`
- (6)控制变量 i 和 j 共同进行循环控制, i 从 1 变到 99, j 从 2 变到 100, 增量均为 2。

```
for (int i=1,j=2; __4_____ ) //i<=99&&j<=100: i+=2,j+=2
```

需要说明的是：可以在 for 循环“控制变量初始化语句”中声明变量，这些变量只在 for 循环结构中有效，离开了该 for 结构，变量就无效了。

3 种循环语句是等效的，但对于循环次数已知的循环操作，用 for 语句实现比较方便；如果是根据某个状态来决定是否要循环下去，用 while 循环比较方便。

## 1[2195] 累加和

输入一个正整数 n，计算 1+2+.....+n

样例输入：10

样例输出：55

<pre>#include &lt;iostream&gt; using namespace std; int main( ) {     int n,i = 1, sum = 0;     cin&gt;&gt;n;     while(i&lt;=n) {         sum = sum+i;         i++;     }     cout&lt;&lt; "sum=" &lt;&lt;sum&lt;&lt;endl;     return 0; }</pre>	<pre>#include&lt;iostream&gt; using namespace std; int main() {     int i,n,sum=0;     cin&gt;&gt;n;     for(_____;_____;_____)         sum=sum+i;     cout&lt;&lt;sum&lt;&lt;endl;     return 0; }</pre>
---	---

### 随堂练习 1

1. 执行循环语句“for(i=1;j<=10;i++) a++;”后变量 i 的值是\_\_\_\_\_。

- A)9                      B)10                      C)11                      D)不确定

2. 以下的 for 循环\_\_\_\_\_。

```
for(x=0,y=0; (y!=123)&&(x<4); x++);
```

- A) 是无限循环              B) 循环次数不定              C) 执行 4 次              D) 执行 3 次

3. 语句 for (表达式 1; ; 表达式 3)等价于? \_\_\_\_\_

- A) for ( 表达式 1; 0; 表达式 3 )  
 B) for ( 表达式 1; 1; 表达式 3 )  
 C) for ( 表达式 1; 表达式 1; 表达式 3 )  
 D) for ( 表达式 1; 表达式 3; 表达式 3 )

4. 以下程序的输出结果是\_\_\_\_\_

- A) 10 1 9 2              B) 9 8 7 6              C) 10 9 9 0              D) 10 10 9 1

```
main( )
{
    int x=10,y=10,i;
```

```

    for(i=0;x>8;y=++i)
        printf( "%d %d ",x--,y);
}

```

## 2[1019]累加求和

累加 100~1000 之内能被 3 整除的偶数。

<pre> #include&lt;iostream&gt; using namespace std; int main() {     int sum=0,i=100;     while(i&lt;=1000){         if((i%3==0)&amp;&amp;(i%2==0))             sum=sum+i;         i++;     }     cout&lt;&lt;sum;     return 0; } </pre>	<pre> #include&lt;iostream&gt; using namespace std; int main() {     int sum=0,i;     for(_____;_____;_____) {         if((i%3==0)&amp;&amp;(i%2==0))             sum=sum+i;     }     cout&lt;&lt;sum;     return 0; } </pre>
---	--

## 3 [2617] 均值

给出一组样本数据，包含  $n$  个浮点数，计算其均值，精确到小数点后 4 位。  
 输入有两行，第一行包含一个整数  $n$  ( $n$  小于 100)，代表样本容量；第二行包含  $n$  个绝对值不超过 1000 的浮点数，代表各个样本数据。  
 输出一行，包含一个浮点数，表示均值，精确到小数点后 4 位。  
 样例输入 2 1.0 3.0  
 样例输出 2.0000

```

int main()
{
    double a,b,x;
    double sum=0,ave; //求和初始值
    int i,n;
    cin>>n;

```

```

    cout.precision(4);
    cout<<fixed<<ave;
    return 0;
}

```

## 4.2 案例分析

### 4 [2198]求最小值

输入 10 个整数，求出其中的最小值。

样例输入：19 63 12 26 94 9 18 34 3 4

样例输出：3

分析：

1. 初始状态是什么\_\_\_\_\_
2. 循环的结束条件时什么\_\_\_\_\_
3. 每次循环做什么事情\_\_\_\_\_

```

#include<iostream>
using namespace std;
int main()
{
    int i,x,min=1000,n;//定义 min 为一个极大值
    for( 1 _____){
        cin>>x;
        2 _____
    }
    cout<<min;
    return 0;
}

```

思考：寻找身边求最小值的例子

### 5 [2620] 最大跨度值

给定一个长度为  $n$  的非负整数序列，请计算序列的最大跨度值(最大跨度值 = 最大值减去最小值)。

一共 2 行，第一行为序列的个数  $n$  ( $1 \leq n \leq 1000$ )，第二行为序列的  $n$  个不超过 1000 的非负整数，整数之间以一个空格分隔。

输出一行，表示序列的最大跨度值。

样例输入

6

3 0 8 7 5 9

样例输出

9

叙述思路: \_\_\_\_\_

```
#include <iostream>
using namespace std;
int main()
{
    int n,x,i,zd=_____,zx=_____;//maxn,minn
    cin>>n;
    for(i=1;i<=n;i++){
        _____
        if(x>zd)_____;
        if(x<zx)_____;
    }
    cout<<zd-zx;
    return 0;
}
```

### 6[1074]计负均正

从键盘输入任意 20 个整型数，统计其中的负数个数并求所有正数的平均值。

保留两位小数

样例输入 1 2 3 4 5 6 7 8 9 10 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10

样例输出

10

5.50

```
#include<iostream>
#include<cmath>
using namespace std;
int main(){
    int i,x,m1=0,m2=0,sum=0;
    double ave;
    for(i=1;i<=20;i++){
        cin>>x;
        if(x<0)m1++;
        if(x>0)
            m2++,sum=sum+x;
    }
    ave=1.0*sum/m2;
```

```
printf("%d\n%.2lf",m1,ave);
}
```

### 4.3 思维拓展：环形路行程问题

在环形道路上的行程问题本质上讲就是追击问题或相遇问题。当二人（或物）同向运动时就是追击问题，追及距离是二人初始距离及环形道路之长的倍数之和；当二人（或物）反向运动时就是相遇问题，相遇距离是二人从出发到相遇所行路程和。

两名运动员在沿湖的环形跑道上练习长跑。甲每分钟跑 250 米，乙每分钟跑 200 米。两人同时同地同向跑，45 分钟后甲追上了乙。如果两人同时同地反向而跑，经过多少分钟后两人相遇？

$$\begin{aligned} & (250-200) \times 45 \div (250+200) \\ &= 50 \times 45 \div 450 \\ &= 2250 \div 450 \\ &= 5 \text{ 分钟} \end{aligned}$$

### 7 [7618] 环形路行程问题

$n$  ( $n \leq 10$ ) 组运动员在沿湖的环形跑道上练习长跑。每组甲乙两名运动员，甲每分钟跑  $a$  米，乙每分钟跑  $b$  米。两人同时同地同向跑， $c$  分钟后甲追上了乙。如果两人同时同地反向而跑，经过  $t$  分钟后两人相遇。问各组运动员相遇的最短时间是多少，保留 1 位小数。

输入

第一行为  $n$ ，表示  $n$  组运动员

第二行为  $a, b, c$

第三行为  $2-n$  组运动员的速度。

输出

最短相遇时间。

样例输入

```
5
250 200 45
260 230
270 200
200 200
300 300
```

样例输出

```
3.8
```

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```

{
    int n,a,b,c,i=1;
    double t,mint=100000;
    int circle;
    cin>>n;
    cin>>a>>b>>c;
    circle=abs(a-b)*c;
    do{
        cin>>a>>b;
        t=1.0*circle/(a+b);
        if(t<mint)
            mint=t;
        i++;
    }while(i<n);
    printf("%.1lf",mint);
    return 0;
}

```

## 课堂作业列表

- |                     |                |
|---------------------|----------------|
| [2195] 累加和          | [1019] 基础累加    |
| [1050] 偶数和          | [2617] 均值      |
| [1073] 查找整除的数       | [2198] 求最小值    |
| [2620] 最大跨度值        | [1074] 计负均正    |
| [2175] 求阶乘          | [7618] 环形路行程问题 |
| [2626] 整数的个数        | [2636] 正常血压    |
| [2637] 统计满足条件的 4 位数 |                |

## 课前练习

1. 以下程序执行后的输出结果是\_\_\_\_\_。

```

#include<iostream>
using namespace std;
int main()
{
    int i,k;
    for(k=0,i=10;i<=100;i+=5 )
        if(i>=60)
            k++;
    cout<<k;
}

```

2. 读程序写结果

```

#include<iostream>

```



```
using namespace std;
int main(){
    int x,y1,y2,y3;
    cin>>x;
    y1=0;y2=1;y3=1;
    while(y2<=x){
        y1=y1+1;y3=y3+2;y2=y2+y3;
    }
    cout<<y1;
}
```

输入：30 输出：\_\_\_\_\_

### 习题

1. 执行循环语句“for(i=1;i<=10;i++)a++;”后变量 i 的值是\_\_\_\_。  
 A)9                      B)10                      C)11                      D)不确定
2. 执行以下语句后， a,b 的值分别是\_\_\_\_\_。  
 for(a=b=0; a<4&&++b; a++) b--;  
 A)3 和 0                      B)3 和 1                      C)4 和 0                      D)4 和 1

3. 有以下程序

```
main( )
{ int i,s=0;
  for(i=1;i<10;i+=2) s+=i;
  printf( “%d\n” ,s);
}
```

程序执行后的输出结果是\_\_\_\_\_

- A) 自然数 1~9 的累加和                      B) 自然数 1~10 的累加和  
 C) 自然数 1~9 中的奇数之和                      D) 自然数 1~10 中的偶数之和

<p>4. 计算 2+4+6+8+...+98+100</p> <pre>int main() {     int i, _____(1)_____;     for(i=2;i&lt;=100;_____(2)_____)         s+=i;     return 0; }</pre>	<p>5. 以下程序的输出结果是_____。</p> <pre>#include &lt;iostream&gt; using namespace std; int main() { int s,i;   for(s=0,i=1;i&lt;3;i++,s+=i);   printf(“%d\n”,s);   return 0; }</pre>
--	--

6. 以下程序的输出结果是\_\_\_\_\_。

```
int main()
{
```

```

        int i=3;
        for( ; i<=18 ;)
        {
            i++;
            if(i%6==1)
                cout<<i<<" ";
        }
    }
}

```

7. 以下程序执行后的输出结果是\_\_\_\_\_

```

#include <iostream>
using namespace std;
int main( )
{
    int i,m=1;
    for(i=5;i>=1;i--)
    {
        m=(m+1)*2;
        printf("m=%d\n",m);
        return 0;
    }
    return 0;
}
输出_____

```

8. 改错：以下程序的功能是计算  $1+1/3+1/5+\dots+1/(2n-1)$  的值，请改正程序中的错误

```

#include<iostream>
1  using namespace std;
2  int main ( )
3  {
4      int n,i;
5      double s=0,t;
6      cin>>n;
7      for (i=1,i<=n,++i)
8          t=1/(2*i-1);
9          s=s+t;
10         cout<<"s="<<s<<endl;
11 }

```

## 5 平均数

重点:

1. 理解平均数的常规求法
2. 掌握用循环求解平均数、最大值、最小值等
3. 强化对循环的理解

### 5.1 知识剖析

在日常生活中，我们常能遇到有关平均数的问题，比如，在排球、篮球等项目的体育比赛中，体育播音员要介绍每名参赛队员的身高，以及每个队的平均身高。我们一听就能了解哪个队队员的身体条件好一些。当然，并不是身体条件好的就一定获胜，但至少这是一种优势。

平均数是一个重要的统计量，应用十分广泛。工农业生产上用平均月产量、平均公顷产量来检验生产效率。用同年龄不同地区儿童的平均身高、平均体重来分析儿童的生长发育的区域差异等等。

平均数应用题的基本特点是，把几个大小不等的数量，在总量不变的情况下，通过移多补少，使它们成为相等的几份，求其中的一份是多少。解题时关键要确定“总数量”以及与“总数量”相对应的“总份数”，然后用总数量除以总份数求出平均数。

求平均数问题的基本数量关系式

$$\text{总数量} \div \text{总份数} = \text{平均数}$$

反过来，已知平均数，我们又可以求出总数量，即

$$\text{总数量} = \text{平均数} \times \text{总份数}$$

在程序中，我们通过循环求出总数量，然后再求平均值。

**例 1:** 气象站在某一天的 1 点、7 点、13 点、19 点测得温度分别是 11 度、14 度、23 度、16 度，算出这一天的平均温度。

解:  $11+14+23+16=64$

$$\text{平均数} = 64 \div 4 = 16$$

#### 1 [2862] 求平均分

已知某班有男同学  $x$  位，女同学  $y$  位， $x$  位男生平均分是 87 分， $y$  位女生的平均分是 85，问全体同学平均分是多少分？输入  $x,y$ ，输出平均分，小数点后面保留 2 位。

样例输入 15 17

样例输出 85.94

```
#include<iostream>
using namespace std;
int main(){
    double a,b,c;
```

```

cin>>a>>b;
c=(87*a+85*b)/(a+b);
cout.precision(2);
cout<<fixed<<c<<endl;
return 0;
}

```

## 2 [1514] 平均分

每次考试之后教师统计学生的成绩和计算平均分都是一件很繁琐的事。不过在现在，计算机的面世大大的降低了工作量。给你一组学生的成绩，要求计算出平均分，精确到小数点后 2 位。

输入每组数据的第一行为一个正整数  $N$ ，表示学生的人数。接下来的  $N$  行，每行为一个正整数，表示各个学生的成绩。

输出对于每一组数据，输出该组学生的平均分，占一行。

样例输入

```

10
90 80 70 65 70 80 64 99 69 78

```

样例输出

```

76.50

```

```

#include<bits/stdc++.h>
using namespace std;
int main ()
{
    int n,a;
    double ave,sum=0;
    ave=0;
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        cin>>a;
        sum+=a;
    }
    ave=sum/n;
    printf("%.2f\n",ave);
    return 0;
}

```

### 随堂练习 1

1. 小于 4 次语文测验的平均成绩是 89 分，第 5 次测验得了 94 分。问：他 5 次测验的平均成绩是多少？

### 3 [7623] 5 班平均分

东方小学五（2）班一小组的期中数学成绩是：得 100 分的 a 人，得 96 分的 b 人，其余 5 人共得了 372 分，第一小组的数学平均成绩是多少分？输入 a,b,输出平均分。

样例输入 3 4

样例输出 88.0

```
#include<bits/stdc++.h>
using namespace std;
int main ()
{
    int a,b;
    double sum=0,ave=0;
    cin>>a>>b;
    ave=_____ ;
    printf("%.1lf",ave);
    return 0;
}
```

### 4 [1312] 十佳歌手大赛

在十佳歌手大赛中，有一组评委给参赛选手打分（百分制）。选手最终得分的获取规则是：去掉一个评委给的最高分，去掉在一个评委给的最低分，取其余分数的平均值。需要说明的是，如果同时有多个评委给出最低分，那么只去掉一个最低分；同样的，如果同时有多个评委给出最高分，那么只去掉一个最高分。你的任务是设计一个程序，自动计算选手的得分。

输入每个测试数据占一行：首先是一个正整数  $N(2 < N < 20)$ ，评委的个数；然后是  $N$  个百分制分数。输出该选手的得分，精确到小数点 2 位有效数字。

样例输入 5 98 90 95 88 85

样例输出 91.00

**分析：** 求出最大值、最小值和总分，总分减去最大值、最小值，然后除以  $n-2$ 。

```
#include <iostream>
using namespace std;
int main()
{
    int n,i,x,_1_____ ;
    double sum=0,ave;
    cin>>n;
    for(i=1;i<=n;i++){
        cin>>_2_____ ;
        if(x>maxn)_3_____ ;
    }
}
```

```

    __4_____;
    sum=__5_____;
}
sum=sum-maxn-minn;
ave=__6_____;
printf("%.2lf",ave);
return 0;
}

```

## 5.2 案例分析

**例 2:** 有一条山路，一辆汽车上山每小时行 30 千米，从原路返回下山时每小时行 60 千米，求汽车上、下山的平均速度。

此题有人这样解： $(30+60)/2=45$ (千米)，即汽车上、下山平均速度是每小时 45 千米。这种解法显然是错误的。

一般来说，求平均速度需要有两个最基本的条件：一是总路程，二是总时间，这题都没。我们不妨假设上山的路程为 60 千米，那么求平均速度就不再困难了

假设上山路程为 60 千米，那么上山时间为  $60 \div 30 = 2$ (时)，下山时间为  $60 \div 60 = 1$ (时)。

根据总路程=总时间一平均速度，可求

汽车上、下山的平均速度为  $(60+60) \div (2+1) = 120 \div 3 = 40$ (千米)

答汽车上、下山的平均速度为每小时行 40 千米

### 5[7624] 春蕾小队爬山

“春蕾小队”5 同学小爬山，每位同学上山的速度是每小时 a 千米，到达山顶后立即沿原路下山。下山的速度是每小时 b 千米。输出每位同学上、下山的平均速度是多少？

输入 5 位同学的上山、下山速度，输出依次输出每位同学上、下山的平均速度。

样例输入	样例输出
3 6	4.0
3 5	3.7
5 7	5.8
4 6	4.8
8 10	8.9

```

#include<bits/stdc++.h>
using namespace std;
int main ()
{
    int i;
    double t1,t2,t,a,b,v;
    for(i=1;_____){
        cin>>a>>b;
        _____
        _____
    }
}

```

```

        printf("%.1f\n",v);
    }
    return 0;
}

```

## 随堂练习 2

1. 甲、乙两地相距 165 千米，一辆汽车从甲地到乙地用了 2.4 小时；沿原路返回时用了 2 小时，求这辆汽车往返甲、乙两地的平均速度？
2. 某班统计期末数学成绩，平均分为 85.13 分。经复查发现李龙的成绩是 87 分，误写成 78 分，经重新计算后，该班数学成绩的平均分是 85.31 分，这个班共有多少名学生？

## 6 [7625] 服装厂

一个服装厂有  $n$  个生产车间，这个月每个车间都计划生成 1000 件，前半个月每个服装厂均只完成了 40% 的生产量，

厂长认为  $n$  个车间生成效率都太低，为了不延误交货日期实现分批交货，厂长规定每个车间剩余的西服分别需要在  $a_i$  天内完成 ( $8 < a_i < 18$ )，问每个车间剩下的日子平均每天至少需要生成多少件西服（数据上取整）？

输入 输入  $n$ ，以及每个车间需要完成的天数

输出 各车间剩下的日子平均每天需要生成多少件西服

样例输入

3

10 16 13

样例输出

60 38 47

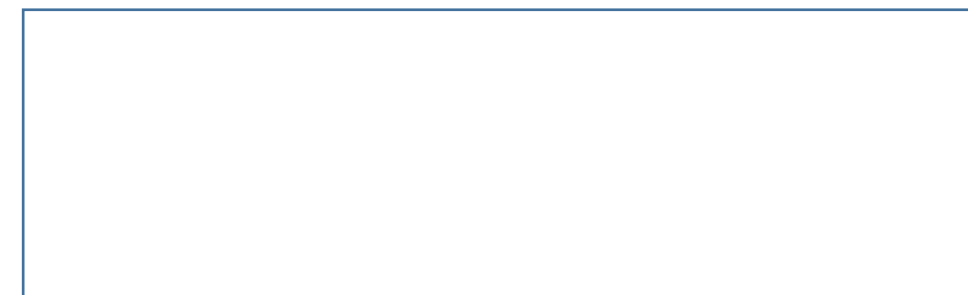
```
int main()
```

```
{
```

```
    int x,i,n;
```

```
    cin>>n;
```

```
    for(i=1;i<=n;i++){
```



```
}
```

```
return 0;
```

}

**例 3:** 有 5 个数，平均数是 10，如果把其中的一个数改为 2，那么这五个数的平均数是 9，这个改动的数原来是多少？

修改后剩余四个数的和为： $5 \times 9 - 2 = 43$

原值 =  $50 - 43 = 7$

## 7[7626] 改动数据

有  $n$  个数，平均数是 10，如果把其中的 1 个数改为 2，那么这  $n$  个数的平均数是  $m$ ，这个改动的数原来是多少？

进行  $p$  次上述操作，输出每次改动的数原来是多少。

样例输入	样例输出
3	7
5 9	8
3 8	10
8 9	

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int i,p,n,m;
```

```
    double x1,x;
```

```
    cin>>n;
```

```
    return 0;
```

```
}
```

## 5.3 思维拓展：数字翻转

给定一个整数，请将该数各个位上数字反转得到一个新数。新数也应满足整数的常见形式，即除非给定的原数为零，否则反转后得到的新数的最高位数字不应为零，例如输入-380，



反转后得到的新数为-83。

输入共 1 行，一个整数 N。  $-1,000,000,000 \leq N \leq 1,000,000,000$ 。

输出共 1 行，一个整数，表示反转后的新数。

样例输入 123

样例输出 321

思考：

1. 求解思路如何？
2. 如何取出每一位数，生成新的数？
3. 负号怎么处理？负数的取余规则如何？

```
#include<iostream>
using namespace std;
int main()
{
    int n,m=0;
    cin>>n;
    while( __1_____)//n!=0
    {
        m= __2_____; //m*10+n%10
        n= __3_____; //n=n/10
    }
    cout<<m<<endl;
    return 0;
}
```

## 课堂作业列表

- |                |                  |
|----------------|------------------|
| [2862] 求平均分    | [1514] 平均分       |
| [7623] 5 班平均分  | [1312] 十佳歌手大赛    |
| [7624] 春蕾小队爬山  | [7625] 服装厂       |
| [7626] 改动数据    | [2224] 输出最大值及其下标 |
| [2642] 求小数的某一位 | [2618] 求整数的和与均值  |
| [2616] 求平均年龄   |                  |

## 课前练习

1. 以下程序执行后的输出结果是\_\_\_\_\_。

```
#include <iostream>
using namespace std;
main( )
{   int i,k;
    for(k=0,i=10;i<=100;i+=5 )
        if(i>=60) k++;
```

```

        cout<<k;
    }

```

2. 以下程序执行后的输出结果是\_\_\_\_\_。

```

#include <iostream>
using namespace std;
main()
{
    int i,s,t;
    s=0;
    for( i=1; i<=10 ; ++i )
    {
        if(i%2==0) t=t-1;
        else t=i;
        s=s+t;
    }
    cout<<"s"<<s;
}

```

## 习题

1. 有如下程序,该程序的执行结果是\_\_\_\_\_

```

#include <iostream>
using namespace std;
main ( )
{
    int i, sum;
    for(i=1;i<=3;sum++) sum+=i;
    cout<<sum;
}

```

A) 6                      B) 3                      C) 死循环                      D) 0

2. 有以下程序:

```

#include <iostream>
using namespace std;
main ( )
{
    int y=6;
    for( ;y>0; y--)
        if(y%3==0)
            cout<<--y;//printf ("%d",--y);
}

```

程序运行后的输出结果是\_\_\_\_\_

A)41                      B)52                      C)63                      D)5421

3. 以下程序执行后 sum 的值是\_\_\_\_\_

A)15                      B)14                      C) 不确定                      D) 0

```

main()
{ int i, sum;
  for(i=1;i<6;i++) sum+=i;
  printf( "%d\n" ,sum);
}

```

4. 有如下程序

```

main ( )
{   int    i,sum;
    for(i=1;i<=3;sum++)  sum+=i;
    printf( "%d\n" ,sum);
}

```

该程序的执行结果是\_\_\_\_\_

A) 6

B) 3

C) 死循环

D) 0

5. 以下程序执行后的输出结果是\_\_\_\_\_

```

#include <iostream>
using namespace std;
int main()
{
  int i,x,a;
  x=10000;
  for(i=1;i<=5;i++){
    scanf("%d",&a);
    if(x>=a) x=a;
  }
  printf("%d",x);
  return 0;
}

```

输入 19 63 12 26 94

输出\_\_\_\_\_

6. 下面程序的功能是求某整数的所有因子之和。请填空。

```

main()
{   int i,x,s=0;
    scanf("%d",&x);
    for( i=1;_____ (1) _____ ;i++ )
        if (x%i==0)
            { s=s+i;_____ (2) _____ ;}
    printf("%d",s);
}

```

7. 输入 n 个整数，求出其中的最小值。

输入：n 个整数；输出：n 个数中的最小值

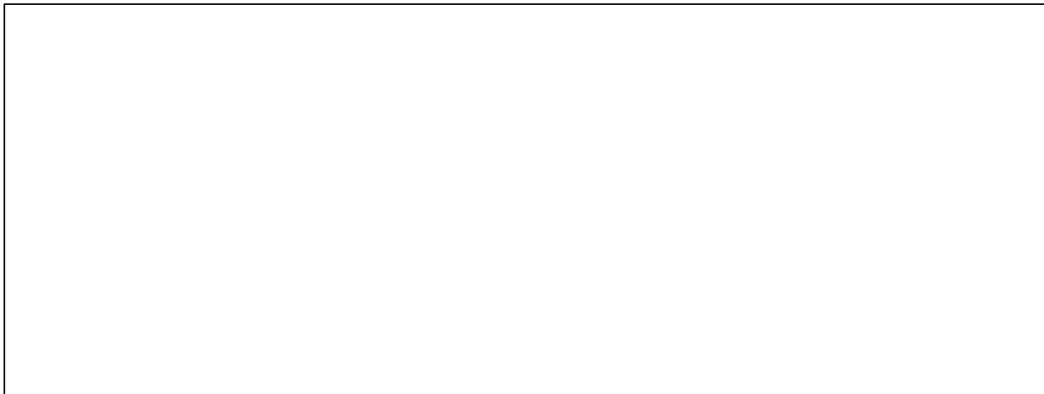
样例输入：

10

19 63 12 26 94 9 18 34 3 4

样例输出：3

```
#include<iostream>
using namespace std;
int main()
{
```



```
    return 0;
}
```

8. 以下程序执行后的输出结果是\_\_\_\_\_

```
#include <iostream>
using namespace std;
int main( )
{   int i,k,sum,m;
    k=sum=0;  m=1;
    for(i=1;i<=10;i++){
        if(i%2)  sum+=i;
        else if(i<=5)  m*=i;
        else  k=k+i;
    }
    printf("%d,%d,%d\n",sum,m,k);
    return 0;
}
```

# 6 循环嵌套

## 内容

1. 理解循环嵌套的含义。
2. 熟练应用循环语句及其嵌套解决一些实际问题。
3. 了解枚举的思想。

循环结构与分支结构的嵌套类似，也可以在一个循环语句的循环体里出现另一个循环语句，不管是 while 语句、do-while 语句还是 for 语句。这样的循环结构称为“循环嵌套”。

```
for(i=1;i<=10;i++){  
    for(j=1;j<=5;j++){  
        ...  
    }  
}
```

(a) for 循环套 for 循环

```
for(i=1;i<=10;i++){  
    while(j<10){  
        ...  
    }  
}
```

(b) for 循环套 while 循环

```
for(i=1;i<=10;i++){  
    for(j=1;j<=5;j++){  
        for(k=1;k<=4;k++){  
            ....  
        }  
    }  
}
```

(c) 多重循环

## 6.1 [1436]阶乘和

求  $S_n=1!+2!+3!+4!+5!+\dots+n!$  之值，  
( $n \leq 10$ )。

样例输入 5

样例输出 153

分析：

1) 求  $i!$  的代码为：

```
int i, t = 1;  
for (j = 1; j <= i; j++) {  
    t = t * j;  
}
```

2) 求和

```
sum = 0;  
for (i = 1; i <= n; i++) {  
    sum = sum + i!  
}
```

```
sum = 0;  
for (i = 1; i <= n; i++) {  
    for (j = 1; j <= i; j++)  
        t = t * j;  
    sum = sum + t;  
}
```

```
# include <iostream>  
using namespace std;  
int main ()  
{  
    int i, j, t, n;  
    int sum = 0;  
    cin >> n;  
    for (i = 1; i <= n; i++) {  
        t = 1;  
        for (j = 1; j <= i; j++)  
            t = t * j;  
        sum = sum + t;  
    }  
    cout << sum << endl;  
    return 0;  
}
```

## 随堂练习 1

1. 以下代码的运行结果是什么？

```
for (i = 1; i <= 5; i++)  
    for (j = 1; j <= 5; j++)  
        cout<<i<<" " <<j;
```

2. 有以下程序

```
int main()  
{  
    int i, j;  
    for(j=1; j<=2; j++)  
    {  
        for(i=3; i>=1; i--)  
            printf("%d ", i+j);  
        printf("\n");  
    }  
    return 0;  
}
```

程序的运行结果是( )

A) 2 3 4

B) 4 3 2

C) 2 3 4

D) 5 4 3

3 4 5

5 4 3

4 3 2

4 3 2

3. 可否把求阶乘和做优化，变成一个循环实现？

## 6.2 完数

完数（Perfect number），又称完美数或完备数，是一些特殊的自然数。它所有的真因子（即除了自身以外的约数）的和（即因子函数），恰好等于它本身。如果一个数恰好等于它的因子之和，则称该数为“完全数”。第一个完全数是 6，第二个完全数是 28，第三个完全数是 496，后面的完全数还有 8128、33550336 等等。

### 2 [7140] 判断完数

输入一个数，判断是否是完数。

样例输入 6

样例输出 yes

分析：

1. 求出除本身外的因子和 sum。

2. 判断 sum 是否等于本身。

```
#include<iostream>
```

```

using namespace std;
int main(){
    int n,sum=0,i;
    cin>>n;
    for( _ 1 _____ )//求因子和
        if( _ 2 _____ )//
    if( _ 3 _____ )//判断是否相等
        cout<<"yes";
    else
        cout<<"no";
    return 0;
}

```

### 3 [2251] 完数

输入两个正整数  $m$  和  $n$  ( $1 \leq m, n \leq 1000$ ), 输出  $m \sim n$  之间的所有完数。完数就是因子和与它自身相等的数 (如  $6=1+2+3$ )。

输入: 输入两个正整数  $m$  和  $n$

输出: 输出所有完数, 输出的数据之间有 2 个空格

样例输入

1

100

样例输出

1 6 28

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int m,n,i,j,sum=0;
    cin>>m>>n;
    if(m==1)//1 特殊处理
        cout<<1;
    for( _ 1 _____ ){
        sum=0;//这里一定赋值为 0 吗?
        for( _ 2 _____ ){
            if(i%j==0)
                sum=sum+j;
        }
        if( _ 3 _____ )
            cout<<" "<<i;
    }
    return 0;
}

```

## 6.3 案例分析

### 4 [2655] 数 1 的个数

给定一个十进制正整数  $n(1 \leq n \leq 10000)$ ，写下从 1 到  $n$  的所有整数，然后数一下其中出现的数字“1”的个数。

例如当  $n=2$  时，写下 1,2。这样只出现了 1 个“1”；当  $n=12$  时，写下 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12。这样出现了 5 个“1”。

输入

正整数  $n$ 。  $1 \leq n \leq 10000$ 。

输出

一个正整数，即“1”的个数。

样例输入

12

样例输出

5

```
#include<stdio.h>
int main(){
    int n,sum=0,i,num;
    scanf("%d",&n);//1321 1 132 2 13 3 1 1 0
    for(i=1;i<=n;i++){
        num=i;
        while(num){
            if(num%10==1)
                sum++;
            num=num/10;
        }
    }
    printf("%d",sum);
    return 0;
}
```

### 5 [2656] 数字统计

请统计某个给定范围  $[L, R]$  的所有整数中，数字 2 出现的次数。

比如给定范围  $[2, 22]$ ，数字 2 在数 2 中出现了 1 次，在数 12 中出现 1 次，在数 20 中出现



1 次，在数 21 中出现 1 次，在数 22 中出现 2 次，所以数字 2 在该范围内一共出现了 6 次。

输入

输入共 1 行，为两个正整数 L 和 R，之间用一个空格隔开。

输出

输出共 1 行，表示数字 2 出现的次数。

样例输入

2 22

样例输出

6

```
#include<iostream>
using namespace std;
int main(){
    int l,r,i,x,sum=0;
    cin>>l>>r;
    for(i=l;i<=r;i++){
        1
        while( 2 ){
            if( 3 )sum++;
            4
        }
    }
    printf("%d",sum);
    return 0;
}
```

## 6.4 思维拓展： Fibonacci 数

斐波那契数列（Fibonacci sequence），又称黄金分割数列、因数学家莱昂纳多·斐波那契（Leonardoda Fibonacci,1175 年-1250 年，中世纪意大利数学家）以兔子繁殖为例子而引入，故又称为“兔子数列”，指的是这样一个数列：0、1、1、2、3、5、8、13、21、34、.....在数学上，斐波那契数列以如下被以递推的方法定义： $F(0)=0$ ， $F(1)=1$ ， $F(n)=F(n-1)+F(n-2)$ （ $n \geq 3$ ， $n \in \mathbf{N}^*$ ）

在现代物理、化学等领域，斐波纳契数列都有直接的应用，为此，美国数学会从 1963 年起出版了以《斐波纳契数列季刊》为名的一份数学杂志，用于专门刊载这方面的研究成果。

斐波那契数列中的斐波那契数会经常出现在我们的眼前——比如松果、凤梨、树叶的排列、某些花朵的花瓣数（典型的有向日葵花瓣），蜂巢，蜻蜓翅膀，等角螺线等。

### 6 [2631] 菲波那契数

题目描述

菲波那契数列是指这样的数列: 数列的第一个和第二个数都为 1, 接下来每个数都等于前面 2 个数之和。给出一个正整数  $k$ , 要求菲波那契数列中第  $k$  个数是多少。

输入

输入一行, 包含一个正整数  $k$ 。 ( $1 \leq k \leq 46$ )

输出

输出一行, 包含一个正整数, 表示菲波那契数列中第  $k$  个数的大小。

样例输入

19

样例输出

4181

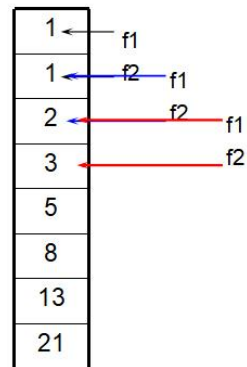
**分析:** 要求 Fibonacci 数列第  $k$  个数, 可以定义个变量表示这  $k$  个数据, 如果看很大, 变量太多, 程序繁琐, 不切实际。而采用递推的方法, 只需要两个变量  $f1$  和  $f2$ , 就可以递推出第  $k$  个数。

$f1$  和  $f2$  表示数列中前后两个数, 初始时,  $f1$  表示第 1 个数,  $f2$  表示第 2 个数。有两种递推方法。

方法一: 每次递推 1 个数

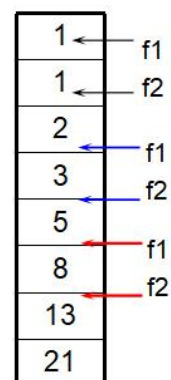
```
#include<iostream>
using namespace std;
```

```
int main( )
{
    //f1,f2 分别代码 Fibonacci 数列中前后两个数
    int f1, f2,k;
    int t;//临时变量
    int i;//循环变量
    f1 = f2 = 1;
    cin>>k;
    for( i=3; i<=k; i++ )
    {
        t = f2;          //(1)先把 f2 保存起来
        f2 = f1+f2;      //(2)f2 是新的项
        f1 = t;          //(3)此时的 f1 是上一次的 f2
    }
    printf( "%d", f2 );
}
```



方法二: 每次递推 2 个数

```
#include<iostream>
using namespace std;
int main( )
```



```

{
    int f1, f2;
    int i,k;
    f1 = f2 = 1;
    cin>>k;
    for( i=1; i<=k/2; i++ )
    {
        f1 = f1+f2;    //(1)左边的 f1 代表第 3 个数，是第 1、2 个数之和
        f2 = f2+f1;    //(2)左边的 f2 代表第 4 个数，是第 2、3 个数之和
    }
    printf("%d",f1);
    return 0;
}

```

## 7 [2321]: 斐波那契数列求和

对于每组输入数据，输出斐波那契数列前 n 项之和。

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    int f1,f2,t,i,sum;
    cin>>n;
    1
    sum=2;
    for(i=3;i<=n;i++)
    {
        t=f2;
        f2= 2
        f1=t;
        sum= 3
    }
    cout<<sum<<endl;
    return 0;
}

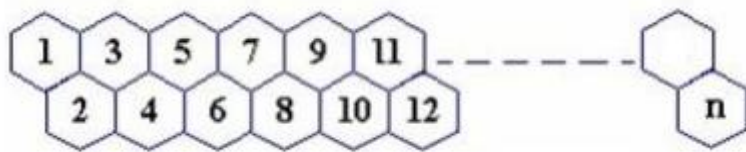
```

## 8[7338] 数字蜂房

题目描述:

一只蜜蜂在下图所示的数字蜂房上爬动，已知它只能从标号小的蜂房爬到标号大的相邻蜂

房，现在问你：蜜蜂从蜂房 M 开始爬到蜂房 N，共有多少种不同的爬行路线？



输入：

只有一行，是 2 个用空格隔开的整数 M，N（ $1 \leq M, N \leq 50$ ）

输出：

只有一行，1 个整数，即有多种爬行路线。

样例输入

1 14

样例输出

377

**题目分析:**

通过观察上图我们可以知道，如果是从 1 开始爬，那么 2 蜂房只能由 1 爬过来，3 可以由 1 或者 2 的位置爬过来。然后我们可以发现从 3 开始，第 n 个蜂房可以由 n-1 或 n-2 蜂房爬过来。由此可得递推式  $f[n]=f[n-1]+f[n-2]$ 。

```
#include<stdio.h>
int main()
{
    long long a,b,f1,f2;
    int i,n;
    scanf("%lld%lld",&a,&b);
    f=0;f1=1;
    for(i=a;i<b;i++)
    {
        f2=f+f1;
        f=f1;
        f1=f2;
    }
    printf("%lld\n",f2);
    return 0;
}
```

## 课堂作业列表

[1436]阶乘和

[7140] 判断完数

[2656] 数字统计

[2321]: 斐波那契数列求和

[2655]数 1 的个数

[2251] 完数

[2631] 菲波那契数

[3292]恰好含有 k 个 3

## 课前练习

1 阅读程序，写出结果\_\_\_\_\_

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    int n,a,m=0;
    cin>>n;
    for(int i=1; i<=n;i++) {
        cin>>a;
        if(a%2==0)m=m+a;
    }
    cout<<m;
}
```

```
2.#include <iostream>
using namespace std;
int main(){
    int n,sum=0,a,i;
    cin>>n;
    for(i=0;i<n;i++)
    {
        cin>>a;
        if(a%2==0)
            sum=sum+a;
    }
    cout<<sum;
    return 0;
}
```

输入：6 2 5 1 6 10 3

输出\_\_\_\_\_

## 课后练习

### 一、选择题

1. 下列程序段执行后 k 值为( )。

```
int k=0,i,j;
for(i=0;i<5;i++)
for(j=0;j<3;j++)
    k=k+1;
```

A) 15

B) 3

C) 5

D) 8

2. 如下程序

```

int i,s,n=0;
for(i=1;i<=5;i++)
{
    s=i%3;
    while(s>0) s--,n++;
}
printf("%d,%d",n,s);
}程序运行后的输出结果是( )

```

A)5,0

B)5,-1

C)6,0

D)6,-1

```

3. int main(){
    int a,b,c,sum=0;
    cin>>a;
    for(int i=1;i<=a;i++)
    {
        b=i/10;
        c=i%10;
        if(b!=7&& c!=7&& i%7!=0)
            sum=sum+i*i;
    }
    cout<<sum<<endl;
    return 0;
}
输入 15
程序运行后的输出结果是_____

```

```

4. #include<iostream>
using namespace std;
int main(){
    int n,i,j,sum,t;
    cin>>n;
    t=1,sum=0;
    for(i=1;i<=n;i++){
        t=t*i;
        sum=sum+t;
    }
    cout<<sum<<endl;
    return 0;
}
输入 5
程序运行后的输出结果是_____

```

5.

```

#include<iostream>
using namespace std;
int main()
{
    int x,y;
    for(x=1;x<5;x=x+2)
    {
        for(y=x;y<4;y++)
            printf("%d ",x+y);
        printf("\n");
    }
}

```

程序的输出结果是： \_\_\_\_\_

6. 输入两个正整数 m 和 n ( $1 \leq m, n \leq 1000$ ), 输出 m~n 之间的所有完数。完数就是因子和与它自身相等的数 (如  $6=1+2+3$ ) 。。

```
int main(){
    int m,n,i,j,sum;
    cin>>m>>n;
    if(m==1)cout<<"1 ";
    for( i=m;i<=n;i++){
        1
        for(2)
            if(i%j==0)3
                if(4)
                    cout<<i<<" ";
    }
    return 0;
}
```

## 7 序列求和

### 重点

1. 强化循环模型的理解。
2. 掌握循环求和问题。

### 7.1 归纳

循环求和的内容可以总结为以下过程：

1. 定义并初始化  $sum, t$
  2. 循环求和，并找出  $a_{n+1}$  和  $a_n$  的递推关系
- ```
for(int i=1;<=n;i++){
    sum=sum+t;//t 表示当前要加的这一项
    找出  $t_{n+1}$  与  $t_n$  的关系
}
```

### 7.2 案例分析

#### 1[2623] 奇数求和

计算非负整数  $m$  到  $n$ (包括  $m$  和  $n$ )之间的所有奇数的和，其中， $m$  不大于  $n$ ，且  $n$  不大于 300。例如  $m=3, n=12$ ，其和则为： $3+5+7+9+11=35$ 。

输入：两个数  $m$  和  $n$ ，两个数以一个空格分开，其中  $0 \leq m \leq n \leq 300$ 。

输出：输出一行，包含一个整数，表示  $m$  到  $n$ （包括  $m$  和  $n$ ）之间的所有奇数的和

样例输入： 7 15

样例输出： 55

分析：遍历所有数字，并判断访问到的当前数是否是奇数即可，如何是就相加。

```
#include<iostream>
using namespace std;
int main()
{
    int sum=0;
    int i,n,m;
    cin>>m>>n;
    for(1_____){
        if 2_____)/
            3_____
    }
}
```



```

cout<<sum;
return 0;
}

```

## 2 [2653] 计算多项式的值

假定多项式的形式为  $x^n+x^{n-1}+\dots+x^2+x+1$ ，请计算给定浮点数  $x$  和正整数  $n$  值的情况下这个多项式的值。 $x$  在 `double` 范围内， $n \leq 1000$ 。多项式的值精确到小数点后两位，保证最终结果在 `double` 范围内。

输入仅一行，包括  $x$  和  $n$ ，用单个空格隔开。 $x$  在 `double` 范围内， $n \leq 1000$ 。

输出一个实数，即多项式的值，精确到小数点后两位。

样例输入 2.0 4

样例输出 31.00

分析：

1. 每一项为  $1, x, x^2, x^3 \dots$  寻找后一项与前一项的关系
2.  $sum=sum+t$  求和加上每一项

```

for(i=1;i<=n;i++){
    求  $x^n$ 
    sum=sum+t;
}

#include <iostream>
using namespace std;
int main()
{
    int n,i;
    double x,item,sum;
    cin>>x>>n;
    1 //初始化

    for(i=1;i<=n;i++){
        sum=2 //求和
        item=3 //递推
    }
    printf("%.2lf",sum);

    return 0;
}

```

### 3 [1021] 前 N 项和

求数列：a, aa, aaa, ..., aa...a 前 N 项和，第 N 项有 N 个 a，a 和 N 均为正整数， $0 \leq a \leq 9$ ， $0 \leq N \leq 9$ ，a 和 N 的值都是从键盘输入的。例如 a=2 时，序列为 2,22,222,2222。。。。。

输入 每组测试数据包含两个整数，分别为 a，n。

输出 每组测试数据输出一行，该行有一个整数，为所求之和。

样例输入 9 6

样例输出 1111104

分析： 递推关系表达式为： \_\_\_\_\_

//前一项和后一项的关系：  $t=t*10+a$ ;

```
#include <iostream>
using namespace std;
int main()
{
    int sum=0,item,a,n,i;
    cin>>a>>n;
    _____ //初始化
    for(i=1;i<=n;i++){
        sum= _____ //求和
        item= _____ //递推
    }
    cout<<sum;

    return 0;
}
```

### 4 [2122]多项式求和 2

观察下面的式子，并找出其中的规律，计算前 n 项和。

$2/1+3/2+5/3+8/5+13/8+21/13.....$

输入每行输入 1 个整数 n ( $0 \leq n \leq 100$ )。

输出每个测试实例的输出占一行，结果保留 3 位小数。

样例输入 13

样例输出 21.334

分析：

递推关系表达式为： \_\_\_\_\_

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    double sum,f1,f2;
```

```

    1
int n,i;
cin>>n;
for(i=1;i<=n;i++){
    double t=f2;
    sum=sum+f2/f1; sum+f2/f1
    2 //更新分子
    3 //更新分母
}
printf("%.3lf",sum);
return 0;
}

```

## 5 [2652] 求出 e 的值

利用公式  $e=1+1/1!+1/2!+1/3!+\dots+1/n!$ ，求 e 的值，要求保留小数点后 10 位。  
 输入只有一行，该行包含一个整数 n ( $2 \leq n \leq 15$ )，表示计算 e 时累加到  $1/n!$ 。  
 输出只有一行，该行包含计算出来的 e 的值，要求打印小数点后 10 位。

样例输入 10

样例输出 2.7182818011

```

#include<iostream>
using namespace std;
int main(){
    int n,i,t;
    double sum;
    1
    cin>>n;
    for(i=1;i<=n;i++){
        2
        3
    }
    cout.precision(10);
    cout<<fixed<<sum<<endl;
    return 0;
}

```

## 6 [1437] 三数的和


求以下三数的和,保留 2 位小数  $1 \sim a$  之和  $1 \sim b$  的平方和  $1 \sim c$  的倒数和  
 输入 a b c

输出  $1+2+\dots+a + 1^2+2^2+\dots+b^2 + 1/1+1/2+\dots+1/c$

样例输入 100 50 10

样例输出 47977.93

```

#include<iostream>
using namespace std;
int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    double sum=0;
    
    cout.precision(2);
    cout<<fixed<<sum<<endl;
}

```

## 7.3 思维拓展：金币

### 7[2660] 金币

国王将金币作为工资，发放给忠诚的骑士。第 1 天，骑士收到一枚金币；之后两天(第 2 天和第 3 天)里，每天收到两枚金币；之后三天(第 4、5、6 天)里，每天收到三枚金币；之后四天(第 7、8、9、10 天)里，每天收到四枚金币.....这种工资发放模式会一直这样延续下去：当连续  $n$  天每天收到  $n$  枚金币后，骑士会在之后的连续  $n+1$  天里，每天收到  $n+1$  枚金币( $n$  为任意正整数)。

你需要编写一个程序，确定从第一天开始的给定天数内，骑士一共获得了多少金币

输入 一个整数（范围 1 到 10000），表示天数。

输出 骑士获得的金币数。

样例输入 6

样例输出 14

**方法一：**找规律，第一个为第一组，后两个为第二组，在后两个为第三组.....以此类推。思路就是找到  $n$  在第几组里，如果  $n$  在这一组的末尾，则直接求平方和；若在这一组的中间，则还需求出它在这一组的第几个，将数据分为从第一组到前一组和这一组来算。判断它在第几组的方法是：发现每一组的最后一个数是所有组数的和，因此将组数升序加起来，一旦当天数  $n$  刚不大于组数，就说明  $n$  在该组里。

**代码：**

```

#include<bits/stdc++.h>
using namespace std;

```

```

int main(){
    int n,a=0,b,sum=0,result=0;
    scanf("%d",&n);//输入天数
    for(int i=1;;i++){ //分为两类讨论
        sum+=i;//循环，直到从第一组开始的每组数据的个数加起来不小于天数
        if(sum>n){//讨论数据个数大于天数的情况，此时 n 在第 i 组里
            for(int j=1;j<=i-1;j++)
                a+=j;//前 (i-1) 组求和得最后一天的天数
            b=n-a;//b 为超出 (i-1) 组的，在第 i 组里的天数
            for(int k=1;k<=i-1;k++)
                result+=pow(k,2);
            result+=b*i;//两部分加起来得结果
            break;
        }
        else if(sum==n){//讨论数据个数小于天数的情况
            for(int k=1;k<=i;k++){
                result+=pow(k,2);
            }//天数正好在第 i 组的末尾，直接求
            break;
        }
    }
    printf("%d",result);//输出结果
    return 0;
}

```

方法二：

```

#include<iostream>
using namespace std;
int main()
{
    int n;//1 2 2 3 3 3 4 4 4 4
    cin >> n; //1+2*2+3*3+4*4+....
    int res = 0;
    int cmp = 1;
    while (n >= cmp)
    {
        n -= cmp;
        res += cmp*cmp;
        cmp++;
    }
    res += n*cmp;
    cout << res << endl;
    return 0;
}

```

方法三:

```
#include<iostream>
using namespace std;
int main()
{
    int K,N,coin=0;
    scanf("%d",&K);
    for(N=1;K-N>=0;K-=N++)
        coin+=N*N;
    printf("%d\n",coin+K*N);
    return 0;
}
```

## 课堂作业列表

[2623] 奇数求和            [2653] 计算多项式的值  
[1021] 前 N 项和        [2122] 多项式求和 2  
[2652] 求出 e 的值        [2639] 计算分数加减表达式的值  
[2660] 金币            [7730] 数字求和  
[1551] 平方和与立方和 J2 [7070] 级数求和  
[3181] 位平方和        [1437] 三数的和

## 课前练习

1. 以下程序的功能是计算:  $s=1+12+123+1234+12345$ 。请填空。

```
#include<iostream>
using namespace std;
int main()
{
    int t=0,s,i;
    _____ (1) _____ ;
    for( i=1; i<=5; i++){
        t=i+_____ (2) _____ ;
        s=s+t;
    }
    printf("s=%d\n",s);
}
```

2.

2 阅读程序, 写出结果\_\_\_\_\_

```
#include <iostream>
using namespace std;
int main()
```

```

{
    int i,n,r,s,x;
    cin>>n;
    for(i=1; i<=n; i++)
    {
        cin>>x;
        s=0;
        while(x!=0)
        {
            r=x%2;
            if(r==1)
                s=s+1;
            x=x/2;
        }
        cout<<s<<<endl;
    }
    return 0;
}

```

输入:

4 2 100 1000 66

输出:

## 课后练习

1 以下程序执行后的输出结果是\_\_\_\_\_

```

int main()
{
    int i,s,t=2;
    s=0;
    for( i=1; i<=6;i++ )
    {
        if(i%2==0) t=t-1;
        else t=-i;
        s=s-t;
    }
    printf("s=%d",s);
}

```

2. #include<stdio.h>

```

int main(){
    int n,i,j=0,a=0;
    double average,sum=0;
    for(i=1;i<=6;i++){
        scanf("%d",&n);
        if(n<0){ j=j+1;}
        if(n>0){

```

```

        sum=sum+n;
        a=a+1;}
    }
    average=sum/a;
    printf("%d\n%.2f\n",j,average);
    return 0;
}

```

输入: 1 -6 5 12 -3 -7

输出\_\_\_\_\_

8. 给定一个序列 2 5 8 11 14..., 输入正整数 n, 求前 n 项和

```

#include <iostream>
using namespace std;
int main(){
    int n,s=2,i,sum=____(1)____ ;
    scanf("%d",&n);
    for(i=2;i<=n;i++){
        s=____(2)____ i
        ____ (3) ____ i
    }
    printf("%d",sum);}

```

4. 求数列的某一项

题目描述

数列的第 1 项为 81, 此后各项均为它前一项的正平方根, 统计并输出该数列前 n 项之和。  
保留小数点后面 6 位精度。

输入 n: 输出前 n 项和

样例输入 5

样例输出 96.048125

```

#include<iostream>
#include<cmath>
using namespace std;
int main()
{
    int i,n;
    double an=81,sum=0;
    cin>>n;
    for(i=1;i<=n;i++)
    {
        sum+=an;
        an=sqrt(an);
    }
}

```



```

cout.precision(6);
cout<<fixed<<sum<<endl;
return 0;
}

```

4. 下面程序的功能是求  $1+(1+2)+(1+2+3)+\dots+(1+2+3+\dots+10)$  的值。

```

main( )
{
    int i,s=0,_____(1)_____ ;
    for (i=1;i<=10;n++)
    {
        _____(2)_____ ;
        s=s+t;
    }
    printf("%d\n",s);
}

```

5. 以下程序的功能是计算  $2-4+6-8+\dots-98+100$  的值，请填空。

```

#include <iostream>
using namespace std;
int main()//
{ int i,t,s,sign=1;
    _____(1)_____ ;
    for (i=2; i<=100; i=i+2)
    { t=sign*i;
      s=_____(2)_____ ;
      _____(3)_____ }
    printf("2-4+6-8+...-98+100=%d\n",s);
}

```

## 8 鸡兔同笼

内容

1. 理解鸡兔同笼的数学原理。
2. 掌握鸡兔同笼的数学原理。
3. 掌握枚举的思想。

### 8.1 知识剖析

“假设”是数学中思考问题的一种方法，有些应用题我们无论是从条件出发用综合法去解答，还是从问题出发用倒推分析法解答，都很难求出答案。但是如果我们合力的进行“假设”，往往能使问题很快得到解决。我国古代趣题“鸡兔同笼”就是运用“假设法”解决问题的一个范例。

1、“鸡兔同笼”问题是：已知笼中鸡、兔共有多少只和脚的总数，求鸡、兔各有多少只。

2、运用“假设法”解题的思路是：

先假设笼子里装的全是鸡（兔），就可以算出在假设下共有多少脚，把这样得到的脚数与题中给出的脚数相比较，看看差多少？每差 2 只脚就说明有一只兔，将所差的脚数除以 2 就可以算出共有多少只兔。

3、解决“鸡兔同笼”问题的基本思路是：

兔数 = (实际脚数 - 每只鸡脚数 × 鸡兔总数) ÷ (每只兔子脚数 - 每只鸡脚数)

思考：在程序中，采用什么方法解决鸡兔同笼问题呢？ 列表或枚举

#### 1 [2200] 鸡兔同笼

题目描述

1500 年前《孙子算经》：今有鸡兔同笼，上有 35 头，下有 94 足，问鸡兔各几何？

现输入头的个数，脚的支数。

输入：输入头的个数，脚的支数。

输出：鸡的个数和兔子的个数

分析：通过列表法或枚举法，遍历所有可能性，看看哪种结果符合要求？

| 鸡的数量 | 兔的数量 | 头 | 脚 | 鸡的数量 | 兔的数量 | 头  | 脚  |
|------|------|---|---|------|------|----|----|
| 0    | 0    | 0 | 0 | 1    | 1    | 2  | 6  |
| 0    | 1    | 1 | 4 | 1    | 2    | 3  | 10 |
| 0    | 2    | 2 | 8 | ...  |      |    |    |
| .... |      |   |   | 23   | 12   | 35 | 94 |
| 1    | 0    | 1 | 2 | ..   |      |    |    |

鸡的数量从 0~35 只: for \_\_\_\_\_  
兔子的数量从 0~35 只: for \_\_\_\_\_  
符合的条件: \_\_\_\_\_

```
#include<iostream>
using namespace std;
int main()
{
    int n1,n2,x,y;//x 表示鸡的只数 y 表示兔
    cin>>n1>>n2;//n1 表示头 n2 表示脚
    for(1 _____)
        for(2 _____)
            if(3 _____)
                cout<<"鸡的个数为"<<x<<" "<<"兔子的个数为"<<y<<endl;
    }
    return 0;
}
```

## 8.2 案例解析

### 2 [3309] 正整数解

题目描述

求  $x^2 + y^2 = n$  的正整数解。

输入  $n$  的值, 输出满足条件的  $x, y$ 。

样例输入: 100

样例输出

6 8

8 6

鸡的数量从 0~35 只: for \_\_\_\_\_

兔子的数量从 0~35 只: for \_\_\_\_\_

符合的条件: \_\_\_\_\_

```
#include<iostream>
using namespace std;
int main( )
{
    int x,y,n;
    cin>>n;
```

```
return 0;
}
```

### 随堂练习 1

1. 王老师用了 117 元买了 18 本书，其中科技书和故事书共 17 本，字典一本（一本字典 17 元）。已知科技书每本 8 元，故事书每本 4 元。问科技书、故事书各买了多少本？

2. 在一个停车场上，停放的车辆（汽车和三轮摩托车）总数恰好是 24。其中辆汽车有 4 个轮子，每辆摩托车有 3 个轮子。这些车共有 86 个轮子。那么，三轮摩托车有多少辆？

**分析：**我们可将汽车“看作兔子”，将三轮摩托车“看作鸡”，轮子“看作腿”，就可用鸡兔同笼的原理来解此题。

**解：**24 辆车如果都算作汽车，那么将有  $24 \times 4 = 96$  (个) 轮子。比现有的 86 个多 10 个轮子。每一辆三轮摩托车比每一辆汽车少一个轮子，故要有 10 辆三轮摩托车来抵消 10 个轮子。

**答：**共有 10 辆三轮摩托车。

**公式套用：**若用基本关系式， $\text{鸡数} = (\text{兔脚数} \times \text{总头数} - \text{总脚数}) \div (\text{兔脚数} - \text{鸡脚数})$

“翻译”为摩托车车辆数计算公式(这里将摩托车看作“鸡”)：

$\text{摩托车数} = (\text{汽车轮子数} \times \text{车辆总数} - \text{轮子总数}) \div (\text{汽车轮子数} - \text{摩托车轮子数})$ ，

即有 摩托车数： $(4 \times 24 - 86) \div (4 - 3) = 10$  (辆)。

### 3 [7632] 三轮摩托车

在一个停车场上，停放的车辆（汽车和三轮摩托车）总数恰好是  $x$ 。其中辆汽车有 4 个轮子，每辆摩托车有 3 个轮子。这些车共有  $y$  个轮子。那么，三轮摩托车有多少辆？

输入 停放的车辆（汽车和三轮摩托车）总数  $x$ ，和  $y$  个轮子。

输出 三轮摩托车的辆数

样例输入 24 86

样例输出 10

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int x,y,i,j;
    cin>>x>>y;
    for( 1 ){}
        for( 2 ){}
            if( 3 )
                cout<<i;
        }
    }
    return 0;
```

```
}
```

改进：用一个循环来实现

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int x,y,i,j;
    cin>>x>>y;
    for(i=0;i<=x;i++){
        j=x-i;
        if(3*i+4*j==y)
            cout<<i;
    }
    return 0;
}
```

#### 4[7633]大小猴子

原始森林里有一只猴王带 35 只(不包括猴王,猴王也不回回摘桃子,只监督,)大小猴子去摘桃子猴王不在的时候,一只小猴子一小时可摘 11 千克桃子,一只大猴子一小时可摘 15 千克桃子。猴王在场监督的时候,每只猴子不论大小每小时都可以多摘 12 千克有一天,他们摘了 8 小时,其中只有第一小时和最后一小时有猴王在场监督,结果共摘了 4400 千克桃子。请你求出这群猴子中,共有多少只小猴子?

#### 分析

这是一道较为复杂的鸡兔同笼类型的题目,除了分成大猴子和小与解猴子两类以外,猴王在场监督和不在场的时候各自的速度均不同但这道题目比较容易的地方在于,猴王监督的总时间是知道的,并且猴王监督时,所有猴子无论大小都是每小时多摘 12 千克,因此我们可以先把这一部分减掉即可转化成一个较为简单的鸡兔同笼问题在减去猴王监督 2 小时,所有的 35 只猴子每小时多摘的 12 千克

以后,我们先假设所有猴子都是小猴子,那还多摘的部分就是把小猴子替换成大猴子多摘的部分。因此,我们可以得到大猴子有

$(4400-2 \times 35 \times 12-35 \times 11 \times 8) \div (15-11) \div 8=15$ (只).

如果:猴王带  $x$  只大小猴子去摘桃子,他们摘了  $z$  小时,其中只有第一小时和最后一小时有猴王在场监督,结果共摘了  $y$  千克桃子。请你求出这群猴子中,共有多少只小猴子?

```
#include<bits/stdc++.h>
using namespace std;
int main(){

    int x,y,z,dhou,xhou;
    cin>>x>>y>>z;
```

```

for( 1 _____){
    xhou 2 _____

    if( 3 _____)
        cout<<xhou;
    }
return 0;
}

```

## 随堂练习 2

### 5 [7740] 蜘蛛蜻蜓

蜘蛛有 8 条腿，蜻蜓有 6 条腿和 2 对翅膀，蝉有 6 条腿和 1 对翅膀，现在这三种小虫共 21 只，有 140 条腿和 24 对翅膀，求每种小虫各几只？

**分析：**此题中出现了 3 种昆虫，不仅有腿的比较，而且又出现了翅膀，显然比例 1 复杂了。解此题的关键就是将 3 种昆虫转化为 2 种昆虫，这样解起来就比较容易了。

突破口在于：蝉和蜻蜓都有 6 条腿。

解：因为蜻蜓和蝉都有 6 条腿，所以从腿的数目考虑，可以把昆虫分成“8 条腿”和“6 条腿”两种，利用基本关系式算出 8 条腿的

$$\text{蜘蛛数} = (140 - 6 \times 21) \div (8 - 6) = (140 - 126) \div 2 = 14 \div 2 = 7(\text{只}).$$

因此，知道了 6 条腿的昆虫共有

$$21 - 7 = 14(\text{只}),$$

也就是蜻蜓和蝉共有 14 只。因为蜻蜓和蝉共有 24 对翅膀，现在再用一次基本关系式，得

$$\begin{aligned} \text{蝉数} &= (14 \times 2 - 24) \div (2 - 1) \\ &= (28 - 24) \div 1 \\ &= 4(\text{只}). \end{aligned}$$

因此，蜻蜓数是  $14 - 4 = 10(\text{只})$ 。

答：有 7 只蜘蛛，4 只蝉，10 只蜻蜓

如果三种小虫共  $x$  只，有  $y$  条腿和  $z$  对翅膀，求每种小虫各几只？

| 蜘蛛 | 蜻蜓 | 蝉     | 翅膀    | 腿     |
|----|----|-------|-------|-------|
| i  | j  | _____ | _____ | _____ |

```

#include<bits/stdc++.h>
using namespace std;
int main(){
    int i,j,k,x,y,z;
    cin>>x>>y>>z;
    for(i=0;i<=x;i++)
        for( 1 _____)
            for( 2 _____){

```

```

        if(3 _____)
            cout<<i<<" "<<j<<" "<<k;
    }
    return 0;
}

```

## 6 [2655]数 1 的个数

给定一个十进制正整数  $n(1 \leq n \leq 10000)$ ，写下从 1 到  $n$  的所有整数，然后数一下其中出现的数字“1”的个数。

例如当  $n=2$  时，写下 1,2。这样只出现了 1 个“1”；当  $n=12$  时，写下 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12。这样出现了 5 个“1”。

输入：正整数  $n$ 。  $1 \leq n \leq 10000$ 。

输出：一个正整数，即“1”的个数。

样例输入：12

样例输出：5

思考：

1. 遍历符合要求的所有的数。
2. 怎么取出每个数中的 1？
3. 初始化有什么特殊要求？

```

#include<iostream>
using namespace std;
int main(){
    int n,num=1,i,K;
    cin>>n;
    for(i=1;i<=n;i++)
    {
        num=i; 这里的 num 作用是什么？可以直接对 i 操作吗？
        while(num!=0)
        {
            if(num%10==1)
                K++;
            num=num/10;
        }
    }
    cout<<K<<endl;
    return 0;
}

```

填空：

```

#include<iostream>

```

```

using namespace std;
int main()
{
    int n,num,   1  
    cin>>n;
    for(int i=1;i<=n;i++)
    {
          2  
        while(  3   )
        {
            if(  4  ) res++;
            num/=10;
        }
    }
    cout<<res<<endl;
    return 0;
}

```

### 8.3 思维拓展：百钱买百鸡

#### 7 [3549]：桐桐去购物

桐桐周末陪妈妈到市场购物。她和妈妈来到一个买鸡的摊位，发现鸡的价格有三种：公鸡每只 5 元钱，母鸡每只 3 元钱，小鸡 3 只 1 元钱。妈妈就给桐桐出了一道计算题：如果用  $n$  元钱买  $m$  只鸡，问公鸡、母鸡和小鸡可以各买多少只？注意：必须把  $n$  元钱正好用完，且买的各种鸡的只数为大于等于 0 的整数。桐桐回到家里便拿起笔来认真计算，算了好久还没得出答案。聪明的你通过编写程序帮助桐桐找出结果好吗？

输入 只有 1 行，两个数  $n$  和  $m$   $0 < n, m \leq 20000$ 。

输出 有若干行，每行有三个数，分别为公鸡、母鸡和小鸡的只数，用空格隔开，按照公鸡只数升序排列。

样例输入

100 100

样例输出

0 25 75

4 18 78

8 11 81

12 4 84

**分析：**如果按照公鸡、母鸡、小鸡全部枚举，循环次数将达到  $10^{15}$ ，远远超过 1 秒（约  $10^{10}$ ），因此需要将 3 个循环，变成两个循环。另外，小鸡 3 只 1 元钱，一次小鸡需满足是 3 的倍数。

```

using namespace std;
int main()

```



```

{
  int gj,mj,xj;
  int m,n;//n 元钱 m 表示数量 输入两个数 n 和 m
  cin>>n>>m;
  for( 1 )
    for(mj 2 ){
      xj= 3
      if( 4 )
        cout<<gj<<" "<<mj<<" "<<xj<<endl;
    }

}

} //head feet

```

### 课堂作业列表

[2200] 鸡兔同笼          [3309] 正整数解  
 [7632] 三轮摩托车       [7633] 大小猴子  
 [7740] 蜘蛛蜻蜓       [3549] 桐桐去购物  
 [2655] 数 1 的个数      6859 楼层编号  
 [3777] 换钞票          [3339] 三连击 2 课前练习

### 课前练习

|                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1. 阅读程序</p> <pre> #include &lt;iostream&gt; using namespace std; int main() {   int x, y, t;   cin &gt;&gt; x &gt;&gt; y;   if (x &lt; y)   {     t = x;     x = y;     y = t;   }   cout &lt;&lt; x &lt;&lt; " " &lt;&lt; y;   return 0; } </pre> <p>输入: 34 52<br/>输出:</p> | <p>2. 阅读程序</p> <pre> #include &lt;iostream&gt; using namespace std; int main() {   int s, x;   x = 0; s = 0;   while (s &lt; 55)   {     x = x + 1;     s = s + x;   }   cout &lt;&lt; x &lt;&lt; endl;   return 0; } </pre> <p>输出:</p> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## 课后练习

1.有以下程序:

```
int main()
{
    int i;
    for(i=5;i<=20; )
    {
        i++;
        if(i<=15&& i%5==0) printf("%d ",i);
    }
}
```

程序运行后的输出结果是( )

A)5 10 15      B)5 10      C)10 15      D)5 10 15 20

2. 程序运行后的输出结果是\_\_\_\_\_

```
int main()
{
    int i,j;
    for(j=2;j<=3;j++)
    {
        for(i=2;i<=j;i++)
        {
            if(j%i==0)break;
            else printf("%d ",i);
            if(i>=j-1) printf("%d ",j);
        }
    }
    return 0;
}
```

3. 程序运行后的输出结果是\_\_\_\_\_

```
#include <iostream>
using namespace std;
int main() {
    int i,j,k;
    for(i=1;i<5;i=i+2)
    {
        k=1;
        for(j=i;j<5;j++)
            k=k+j;
    }
    printf("s=%d\n",k);
}
```

4.以下程序的输出结果为\_\_\_\_\_

```
int main()
{
    int i,j,s;
    for(i=6;i<=100;i++)
    {
        s=1;
        for(j=2;j<=i/2;j++)
            if(i%j==0) s+=j;
        if(s==i)
        {
            printf("%d=1",i);
            for(j=2;j<=i/2;j++)
                if(i%j==0) printf("+%d",j);
            printf("\n");
        }
    }
}
```

5. 求满足下式的  $x,y,z$

```

      X Y Z
+   Y Z Z
-----
      5 3 2

```

```

int main()
{ int x,y,z,i,result=532;
  for (x=1; ___(1)___ ;x++)
    for (y=1; ___(2)___ ;y++)
      for ( ___(3)___ ; ___(4)___ ;z++)
        { i= ___(5)___ +(100*y+10*z+z);
          if(i==result){
            printf("x=%d",x);
            printf("y=%d",y);
            printf("z=%d\n",z);
          }
        }
}

```

## 9 break/continue

内容:

1. Break、continue 的用法。
2. 理解状态及状态变量的概念。

在循环结构中，有时需要提前跳出循环体，或者忽略本次循环的后续语句而去执行下一次循环。为此，C++ 提供了 `break` 语句和 `continue` 语句。

### 9.1 break 语句

如果循环条件永远为真，这种循环称为永真循环，如下面的两个例子。当然，这种循环应该在适当时候用 `break` 语句退出循环。

```
while( 1 )
{
    ...
}
while( true )
{
    ...
}
```

`break` 语句:

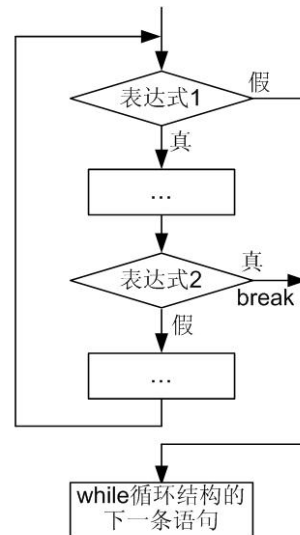
- 可以在 `switch` 结构中使用;
- 还可以用于循环体内。其作用为：使流程从循环体内跳出循环体，即提前结束循环，接着执行整个循环语句后面的语句。

`break` 语句只能用于循环语句和 `switch` 语句内，不能单独使用或用于其他语句中。

```
while(表达式 1)
{
    ...
    if(表达式 2) break;
    ...
}
```

思考，输出以下程序的结果:

```
1.
#include<iostream>
using namespace std;
int main( )
{
    int i,j,k;
    for(j=1;j<=5;j++){
```



```

        cout<<i<<" "<<j<<endl;
        if(j==3){
            break;//退出第一重循环
        }
    }
}

```

结果为: \_\_\_\_\_

2.

```

#include<iostream>
using namespace std;
int main( )
{
    int i,j,k;
    for(i=1;i<=5;i++)
        for(j=1;j<=5;j++){
            cout<<i<<" "<<j<<endl;
            if(j==3){
                break;//退出第一重循环
            }
        }
}

```

结果为: \_\_\_\_\_

## 1[6854]与 7 无关的数

一个正整数，如果它能被 7 整除，或者它的某一位上的数字为 7，则称其为“与 7 相关”的数。现在，请编程求出所有小于或等于  $n$  的“与 7 无关”的正整数个数。

输入一行一个正整数  $n$ ， $n \leq 10^6$ 。

输出一行一个整数，表示答案。

样例输入 21

样例输出 17

### 问题分析

判断  $x$  中是否有数字 7 的时候，一旦出现“ $x \% 10 == 7$ ”，则说明  $x$  是与 7 相关的数，就可以用 `break` 语句立刻退出当前的 `while` 循环，这样可以提高程序的效率。

```

#include<iostream>
using namespace std;
int main(){
    int n,ans = 0;
    cin >> n;
    for(int i = 1; i <= n; i++){

```

```

        int flag1 = 1,flag2 = 1;
        if(i % 7 == 0) flag1 = 0;
        int x = i;
        while(x){
            if(x % 10 == 7) {flag2 = 0; break;}
            x = x / 10;
        }
        if(flag1 && flag2) ans++;
    }
    cout << ans << endl;
    return 0;
}

```

## 2 [1048] 统计输出人数

编写程序，输入一批学生的成绩，遇 0 或负数则输入结束，要求统计并输出优秀（大于等于 85）、通过（60~84）和不及格（小于 60）的学生人数。

样例输入

88 71 68 70 59 81 91 42 66 77 83 0

样例输出

>=85:2

60-84:7

<60:2

分析：遇 0 或负数则输入结束，如何表达？ \_\_\_\_\_

```

#include<iostream>
using namespace std;
int main(){
    int i,x1=0,x2=0,x3=0,n;
    while(1){
        cin>>n;
        if(n<=0)
            break;
        if(n>=85) x1++;
        if((n>=60)&&(n<=84))x2++;
        if(n<60) x3++;
    }
    cout<<">=85:"<<x1<<endl;
    cout<<"60-84:"<<x2<<endl;
    cout<<"<60:"<<x3<<endl;
    return 0;
}

```

### 3 [2121] 多项式求和

观察下面的式子，并找出其中的规律，计算前  $n$  项和。

$$1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + \dots$$

输入 每行输入 1 个整数  $n$  ( $0 \leq n \leq 100$ )，当  $n=0$  时，表示输入结束，不做运算。

输出 每个测试实例的输出占一行，结果保留 3 位小数。

样例输入

1

2

3

0

样例输出

1.000

0.500

0.833

分析：

1. 如何实现多组数据输入？

```
while(cin>>n){}  
while(scanf("%d",&n)!=EOF){}
```

2. 当  $n=0$  时，表示输入结束

```
if(n==0)break;
```

3. 循环语句是什么？

```
for(int i=1;i<=n;i++){  
    sum=sum+1.0*f/i;  
    f=-f;  
}
```

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n,f;
```

```
    double sum;
```

```
    while(cin>>n){
```

```
        1 _____
```

```
        sum=0;
```

```
        2 _____
```

```
        for(int i=1;i<=n;i++){
```

```
            sum= 3 _____
```

```
            4 _____
```

```
        }
```

```
        printf("%.3f\n",sum);
```

```

    }
    return 0;
}

```

#### 4 [1481]数列求和

题目描述

求数列 1、2、3、2、3、4、3、4、5、4、5、6、...前 N 项和。

输入

输入文件中包含多个测试数据。每个测试数据占一行，为一个自然数 N， $1 \leq N \leq 1000$ 。输入文件最后一行为 0，表示输入结束。

输出

对输入文件中的每个测试数据 N，输出数列前 N 项和。

样例输入

10

21

0

样例输出

31

105

分析：循环语句是什么？

a. 怎么实现 3 个 3 个递推？

```

    for(j=i;j<i+3;j++){//i=1,2,3...
        k++;
        sum=sum+j;
    }

```

b. 怎么判断已经累加结束？

```

    if(k==n){
        cout<<sum<<endl;
        break;
    }

```

方法 1:

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n,sum,i,j,k;
```

```
    while(cin>>n){
```

```
        if(n==0)break;
```

```
        sum=0,k=0; 1 k 的含义是什么？
```

```
        for(i=1;i<=1001;i++){ 2 这个循环什么意思？
```

```
            for(j=i;j<i+3;j++){
```

```
                k++;
```

```
                sum=sum+j;
```

```
                if(k==n){ 3
```

```
                    cout<<sum<<endl;
```



```

        break;
    }
}
if(k>=n)break; 4
}
}
return 0;
}

```

方法 2:

```

#include<iostream>
using namespace std;

int main(){
    int n,i,j,sum,a,b,c;
    while(cin>>n){//1 2 3 2 3 4
        if(n==0)break;
        sum=0;//求和初始值
        c=6;//第一轮的和
        b=n/3;//需要多少轮 3 个数为一轮
        a=n%3;//多出的数特殊处理
        for(i=1;i<=b;i++){
            sum=sum+c;
            c=c+3;//每一轮在前一轮的基础上+3
        }
        for(j=1;j<=a;j++){
            b=b+1;//加上剩余的数
            sum=sum+b;
        }
        cout<<sum<<endl;
    }
    return 0;
}

```

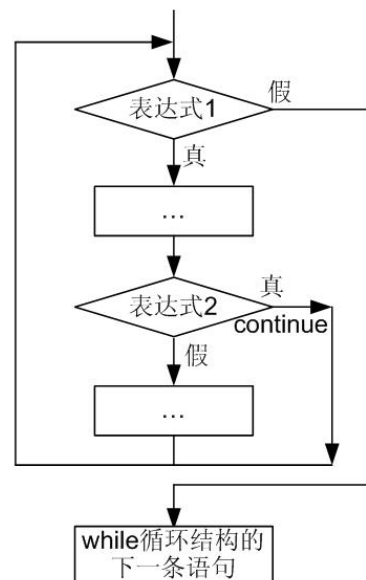
## 9.2 continue 语句

continue 语句只能用于循环结构内，其作用为：结束本次循环，即跳过循环体中下面尚未执行的语句，接着进行下一次是否执行循环的判定。

```

while(表达式 1)
{
    ...
    if(表达式 2) continue;
    ...
}

```



```

}

#include<iostream>
using namespace std;
int main( )
{
    int i,j,k;
    for(i=1;i<=5;i++)
        for(j=1;j<=5;j++)
            if(j==3){
                cout<<i<<" "<<j<<endl;
                continue;
            }
}

```

## 5 [2654] 与 7 无关的数

### 【问题描述】

一个正整数，如果它能被 7 整除，或者它的某一位上的数字为 7，则称其为“与 7 相关”的数。

请编程求出所有小于或等于  $n$  的“与 7 无关”的正整数个数。

输入一行一个正整数  $n$ ， $n \leq 10^6$ 。

输出一行一个整数，表示答案。

样例输入 21

样例输出 17

### 问题分析

当  $i$  是 7 的倍数，即“ $i \% 7 == 0$ ”时，说明  $i$  已经是与 7 相关的数了，此时，就不再需要判断  $i$  的某一位是否是 7 了。所以，可以用 **continue** 忽略本次循环的后续语句，而去执行下一次循环，判断下一个数了。这样做同样可以**提高程序的效率**。

```

#include<iostream>
using namespace std;
int main(){
    int n,ans = 0;
    cin >> n;
    for(int i = 1; i <= n; i++){
        int flag = 1;
        if(i % 7 == 0) continue;
        int x = i;
        while(x){
            if(x % 10 == 7) {flag = 0; break;}
            x = x / 10;
        }
        if(flag) ans++;
    }
    cout << ans << endl;
}

```

```

        }
        if(flag) ans++;
    }
    cout << ans << endl;
    return 0;
}

```

## 6 [3324] 小鱼的航程

有一只小鱼，它上午游泳 150 公里，下午游泳 100 公里，晚上和周末都休息（实行双休日），假设从周  $x(1 \leq x \leq 7)$  开始算起，请问这样过了  $n$  天以后，小鱼一共累计游泳了多少公里呢？

输入两个整数  $x, n$  (表示从周  $x$  算起，经过  $n$  天， $n$  在 `long int` 范围内)。

输出一个整数，表示小鱼累计游泳了多少公里。

样例输入 3 10

样例输出 2000

**思考：**

1. 如何表达晚上和周末都休息？ \_\_\_\_\_

```

#include<iostream>
using namespace std;

int main(){
    int n, x, ans=0;
    cin>>x>>n;
    for(int i=x; i<=x+n-1; i++)
        if((i%7)==6 || (i%7)==0)
            continue;
        else
            ans+=250;
    printf("%d", ans);
}

```

## 9.3 思维拓展：素数

质数，又称素数，是指在大于 1 的自然数中，除了 1 和它本身以外不再有其他因数的自然数。

### 2 [3199] 素数判断

判断某个大于 2 的正整数是否为素数。

注意：是素数输出 yes，不是素数输出 no，其中 yes 和 no 均为小写。

样例输入

5

样例输出

Yes

```
#include <iostream>
using namespace std;
int main() {
    int n,i;
    int flag=1;//默认为素数
    cin>>n;
    for(i=2;i<n;i++){
        if(n%i==0){
            flag=0;
            break;
        }
    }
    if(flag==1)
        cout<<"yes"<<endl;
    else cout<<"no"<<endl;
    return 0;
}
```

## 课堂作业列表

|                 |               |
|-----------------|---------------|
| [6854] 与 7 无关的数 | [1048] 统计输出人数 |
| [2121] 多项式求和    | [1481] 数列求和   |
| [3324] 小鱼的航程    | [3199] 素数判断   |
| [7332] 最佳时间     |               |

## 课前练习

### 1 阅读程序

```
#include <iostream>
using namespace std;
int main()
{
    long x;
    int i, n;
    cin >> n;
    x = 1;
    for (i = 1; i <= n; i++)
        x = x * i;
    cout << x;
```

```
    return 0;
}
```

输入:

8

输出:

## 2 阅读程序

```
#include <iostream>
using namespace std;
int main()
{
    int x, y, t;
    cin >> x >> y;
    if (x > y)
    {
        t = x;
        x = y;
        y = t;
    }
    t = 0;
    while (y >= 0)
    {
        y = y - x;
        t = t + 1;
    }
    cout << "t=" << t - 1;
    return 0;
}
```

输入:

512 16

输出:

## 课后练习

1. 有以下程序段:

```
int k=1;
while(10)
{    k++;
    if(k) break;
```

} 则下面描述中正确的是( )

- A) 1次也不执行循环体语句
- C) 执行 10 次循环体语句

- B) 执行 1 次循环体语句
- D) 有语法错

2.程序运行结果是\_\_\_\_\_

```
#include<iostream>
using namespace std;
int main()
{
    int i,j,a=0;
    for( i=1; i<=2; i++)
    {
        a++;
        for( j=0; j<=3; j++)
        {
            if(j%3) break;
            a++;
        }
        a=a+j;
    }
    printf("%d\n",a);
}
```

3.程序运行结果是\_\_\_\_\_

```
#include <iostream>
using namespace std;
main( )
{
    int i;
    for( i=3; i>=0; i--)
    {
        if(i%3) printf("@");
        else continue;
        printf("$");
    }
    printf("&\n");
}
```

4.

```
main( )
{
    int i,j,a=0,b=0;
    for( i=1;i<=2; i++)
    {
        for (j=1;j<=3;j++)
        {
            if( j%2==0) break;
            a++;
        }
        b=b+a++;
    }
    printf("%d\n",a);
}
```

程序运行后的输出结果是\_\_\_\_\_。

A)4      B)5      C)6      D)8

5.

```
main( )
{
    int i,a=1;
    for( i=1; i<=10; i++)
    {
        if(a>=10) break;
        if(a%3==1) a=a+2;
        else a=a+i;
    }
    printf("%d\n",i);
}
```

```
}
```

程序运行后的输出结果是\_\_\_\_\_。

A)5      B)6      C)7      D)8

6. 有以下程序:

```
#include<iostream>
using namespace std;
int main( )
{
    int i=1,j=9,n=0;
    while(i<j)
    {
        n=n+i+j;
        if(n>=30)
            break;
        i++; j--;
        n=n+i;
    }
    printf("%d\n",n);
}
```

程序运行后的输出结果是\_\_\_\_\_。

A)0      B)30      C)35      D)40

# 10 质数

内容:

1. 掌握状态变量的用法。
2. 掌握质数的求解及其变化。

## 10.1 知识剖析

质数，又称素数，是指在大于 1 的自然数中，除了 1 和它本身以外不再有其他数整除的自然数。例如 11，它不能被除了 1 和它本身以外的数整除，所以 11 是素数。

如何判断一个数是素数呢？假设自然数  $m$ ，在判断  $m$  是否为素数的过程中，要判断 2 是否能整除  $m$ ，如果能整除则  $m$  是合数，要提前退出循环，不再继续判断；否则继续判断 3 是否能整除  $m$ ，如果能整除则  $m$  是合数，要提前退出循环，不再继续判断；...；一直判断到小于  $m$  的最大整数为止。如果在这个过程中始终没有找到能整除  $m$  的数，则  $m$  是素数。

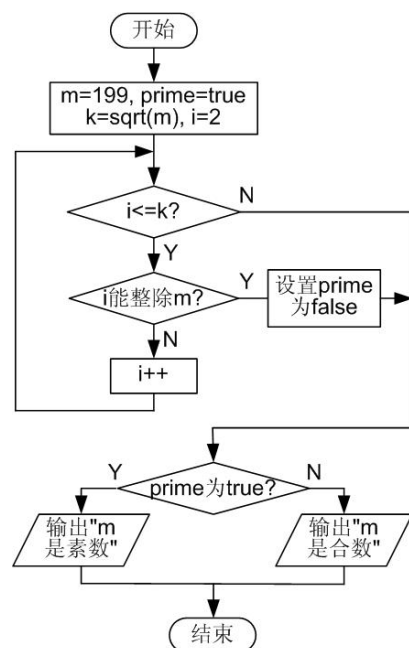
循环结构执行完毕后，如何判断  $m$  是否为素数，有两种方法。

### 方法一(使用状态变量)

用来标志某种状态的变量，通常为 `bool` 型，也可以为 `int` 型)：

1. 在程序中定义一个状态变量 `prime`，其初值为 1，默认该数为素数；
2. 当找到第一个能整除  $m$  的  $i$  时，设置 `prime` 为 0，并提前退出循环；
3. 如果始终没有找到能整除  $m$  的  $i$ ，则状态变量 `prime` 的值仍为初值 1。
4. 当 `for` 循环执行完毕后，根据 `prime` 的值可判定  $m$  是否为素数。

```
#include <iostream>
using namespace std;
int main( )
{
    int m, i, k;
    int prime = 1; //状态变量
    cin>>m;
    for( i=2; i<m; i++ )
    {
        //找到第一个能整除 m 的 i
        if( m%i==0 )
        {
            prime = 0;//设置 prime 为 0
            break;//并退出循环
        }
    }
    if( prime ) printf( "%d 是素数\n", m );
    else printf( "%d 是合数\n", m );
}
```



(a) 使用状态变量



```
}
```

思考：在求素数的遍历过程中，循环右边界需要到  $m-1$  吗？ `for( i=2; i<m; i++)`

改进的写法：

```
for(i=2;i<=sqrt(m);i++)  
for(i=2;i*i<=m;i++)
```

### 方法二(不使用状态变量)

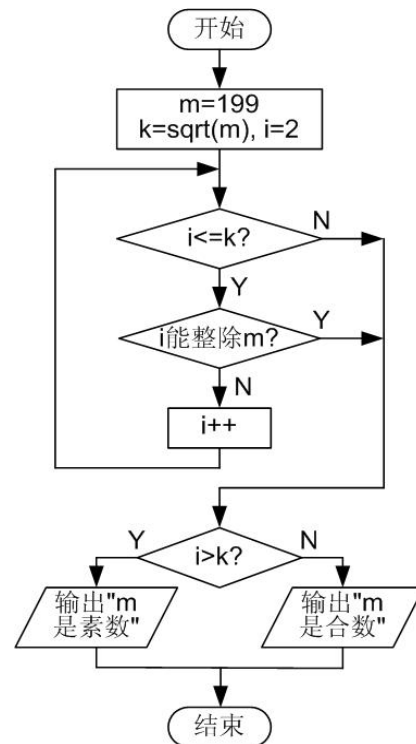
根据 `for` 循环执行完后循环变量的值及循环变量的终值的关系来判定  $m$  是否为素数。

该循环有 2 个出口：

一是在判断  $m$  是否为素数的过程中，已找到一个能整除  $m$  的整数  $i$ ，要提前退出循环，这样循环变量  $i$  的值是  $\leq k$  的；( $k$  的值是什么？)

二是判断到  $k$  都没找到能整除  $m$  的整数  $i$ ，这时循环条件不再成立，要退出循环，此时循环变量  $i$  的值是  $k+1$

```
#include <iostream>
using namespace std;
int main()
{
    int m, i, k;
    cin >> m;
    k = (int)sqrt(m);
    for( i=2; i<=k; i++ )
    {
        if( m%i==0 ) break;
    }
    if( i>k ) printf( "%d 是素数\n", m );
    else printf( "%d 是合数\n", m );
    return 0;
}
```



(b) 不使用状态变量

### 方法三：

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n,j=2;
    cin >> n;
    while(j<=sqrt(n)&& n%j!=0)j++; 谈谈这句话的意思?
    if(j>sqrt(n))
    cout<<"yes"<<endl;
    else
    cout<<"no"<<endl;
    return 0;
}
```

## 1 [3199] 素数判断

判断某个大于 2 的正整数是否为素数, 是素数输出 yes, 不是素数输出 no.

```
#include <iostream>
using namespace std;
int main() {
    int n,i;
    int flag=_____1;//默认为素数
    cin>>n;
    for(1_____){
        if(2_____){
            flag=3_____
            4_____
        }
    }
    if(flag==1)
        cout<<"yes"<<endl;
    else cout<<"no"<<endl;
    return 0;
}
```

## 10.2 案例解析

### 2 [2253]素数的个数与和

输入两个正整数  $m$  和  $n$  ( $1 \leq m, n \leq 500$ ), 统计并输出  $m$  和  $n$  之间的素数的个数以及这些素数的和, 素数就是只能被 1 和自身整除的正整数, 1 不是素数, 2 是素数。

样例输入 1 10

样例输出 4 17

```
#include <iostream>
#include<cmath>
using namespace std;

int main()
{
    int m,n,i,j,flag,k=0,sum=0;
    cin>>m>>n;
    if(m==1)m=2;//1 特殊处理
    for(i=m;i<=n;i++){
        1_____//表示初始值默认都是素数
```

```

for(2 _____){
    if(3 _____){
        4 _____
        5 _____
    }
}
if(flag==1){
    k++;
    6 _____ sum=sum+i;
}
}
cout<<k<<" "<<sum;
return 0;
}

```

### 3 [3304] 素数-小数据

计算出所有小于等于给定正整数的素数个数(我们认为 1 不是素数)

输入文件中包含多个测试数据。每个测试数据占一行，为一个正整数  $N(0 < N \leq 1000)$ ，即给定正整数。测试数据一直到文件尾。

对输入文件中的每个正整数，输出所有小于等于给定它的素数个数。

| 样例输入 | 样例输出 |
|------|------|
| 2    | 1    |
| 3    | 2    |
| 4    | 2    |
| 5    | 3    |
| 9    | 4    |

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n,i,flag,k;//k 用于计数
    while(cin>>n){

```

```

        cout<<k<<<endl;
    }
    return 0;
}

k=0;
for(int i=2;i<=n;i++){
    flag=1;
    for(int j=2;j<i;j++){
        if(i%j==0){
            flag=0;
            break;
        }
    }
    if(flag==1)
        k++;
}

```

#### 4 [7631] 素数-大数据

你的任务是计算出所有小于等于给定正整数的素数个数(我们认为 1 不是素数)  
 输入 每个测试数据占一行, 为一个正整数  $N(0 < N \leq 100000)$ , 即给定正整数。  
 输出 输出所有小于等于给定它的素数个数。  
 样例输入 888888  
 样例输出 70638

**分析:** 当数据比较大时, 如  $n=10^6$ , 两个 for 循环, 运行的次数将达到  $10^{12} \gg 10^{10}$ , 超过正常 1 秒钟的运行时间, 这是需要降低时间复杂度, 采用:

```

for(int i=2;i<=n;i++)
    for(int j=2;j*i<=n;j++) 来处理, 此时的运行的次数为  $10^9$ 

```

#### 5[2658] 质因数分解

已知正整数  $n$  是两个不同的质数的乘积, 试求出较大的那个质数。

输入只有一行, 包含一个正整数  $n$ 。  
 对于 60% 的数据,  $6 \leq n \leq 1000$   
 对于 100% 的数据,  $6 \leq n \leq 2 \times 10^9$   
 输出只有一行, 包含一个正整数  $p$ , 即较大的那个质数。  
 样例输入 21  
 样例输出 7

**思路:** \_\_\_\_\_

```

#include<iostream>
using namespace std;
int main(){
    int n,i,j;
    scanf("%d",&n);
    for(1){
        if(n%i==0){
            j= 2
            printf("%d",3);j
            break;
        }
    }
    return 0;
}

```

## 6 [3556] 孪生素数

桐桐把大小之差不超过 2 的两个素数称为一对孪生素数，如 2 和 3、3 和 5、17 和 19 等等。请你帮助桐桐统计一下，在不大于自然数  $N$  的素数中，孪生素数的对数。

输入一个自然数  $N(N \geq 4 \& \& N \leq 10^6)$ 。

输出一个整数，表示  $N$  以内孪生素数的对数。

样例输入 20

样例输出 5

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
    int ans=0;
    int flag1,flag2;
    for(int i=2;i<=n-2;i++)
    {
        if(i==2){
            ans++;
            continue;
        }
        flag1=1,flag2=1;
        for(int j=2;j*j<=i;j++)
        {
            if(i%j==0) {
                flag1=0;
                break;
            }
        }
        if(flag1==1) ans++;
    }
}

```

```

        }
    }
    for(int j=2;j*j<=i+2;j++)
    {
        if((i+2)%j==0) {
            flag2=0;
            break;
        }
    }
    if(flag1&&flag2)
        ans++;
}
cout<<ans<<endl;

return 0;
}

```

**思考:** 上述代码, 在判断  $i$  和  $i+2$  是否素数的代码基本相同, 可否写一次, 然后重复使用呢?

## 9.4 思维拓展 函数

“函数”这个名词的英文原文是“function”，而“function”的原意是“功能”。顾名思义，函数就是用来实现某个功能的，而且通常只实现一个功能。也就是说，在程序设计语言里引入函数的概念，就是为了进行功能分解和代码重用。

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int isprime(int x)
```

```
{
```

```
    for(int i=2;i*i<=x;i++)
```

```
    {
```

```
        if(x%i==0) return 0;
```

```
    }
```

```
    return 1;
```

```
}
```

```
int main()
```

```
{
```

```
    int n;
```

```
    cin>>n;
```

```
    int ans=0;
```

```
    for(int i=2;i<=n-2;i++)
```

```
    {
```

```
        if(i==2)
```

```

    {
        ans++;
        continue;
    }
    if(isprime(i)==1&&isprime(i+2)==1)
    {
        ans++;
    }
}
cout<<ans<<endl;
return 0;
}

```

## 课堂作业列表

- |                 |                 |
|-----------------|-----------------|
| [3199] 素数判断     | [2253]素数的个数与和   |
| [3304] 素数-小数据   | [7631] 素数-大数据   |
| [2659] 第 n 小的质数 | [2658] 质因数分解    |
| [3556] 孪生素数     | [2020] 哥德巴赫猜想   |
| [2371] 又是素数!    | [2358]素数对猜想课前练习 |

## 课前练习

1. 以下程序运行输出后的结果是\_\_\_\_\_

```

main()
{   int n=12345,sum=0;
    while(n)
    {   sum+=n%10;
        n=n/10;
    }
    printf("%d",sum)
}

```

2.以下程序运行输出后的结果是\_\_\_\_\_

```

#include<iostream>
using namespace std;
int main()
{
    int i,j,a=0;
    for( i=1; i<=2; i++)
    {
        a++;
        for( j=0; j<=3; j++)
        {   if(j%3)   break;

```

```

        a++;
    }
    a=a+j;
}
printf("%d\n",a);
return 0;
}

```

## 习题

1.有以下程序:

```

main( )
{
    int i,a=1;
    for( i=1; i<=10; i++)
    {
        if(a>=10) break;
        if(a%3==1) a=a+2;
        else a=a+i;
    }
    printf("%d\n",i);
}

```

程序运行后的输出结果是\_\_\_\_\_。

A)5    B)6    C)7    D)8

2. 执行以下程序后的输出是(                    )

```

int main()
{
    int x,y;
    for(x=1,y=1;x<=20;x++)
    {
        if(y>20)break;
        if(y%5==1){y=y+5;continue; }
        y=y-1;
    }
    printf("x=%d,y=%d\n",x,y);
}

```

A)x=5,y=21

B)x=4,y=21

C)x=5,y=20

D)x=4,y=20

3.

```

main( )
{
    int i;
    for( i=3; i>=0; i--)
    {
        if(i%3) printf("@");
        else continue;
        printf("$");
    }
}

```



```

        printf("&\n");
    }

```

程序运行后的输出结果是\_\_\_\_\_。

A) @\$#\$      B) @\$#\$&      C) @\$@\$&      D) @@&

4. 执行以下程序后的输出是(                      )      3

```

#include<iostream>
using namespace std;
int main()
{
    int i,j,a=0,b=0;
    for( i=1;i<=2; i++)
    {
        for(j=1;j<=3;j++)
        {
            if( j%2==0 ) break;
            a++;
        }
        b=b+a;
    }
    printf("%d\n",b);
}

```

5. 输入 12 输出\_\_\_\_\_

```

int main()
{
    int a,b,i,j;
    scanf("%d",&a);
    b=0;
    for(j=1;j<=15;j++)
    {
        for(i=2;i<=a;i++)
        {
            if(a%i==0)
            {
                b+=i;
                a=a/i;break;
            }
        }
        if(a==1)
            break;
    }
    printf("%d\n",b);
    return 0;}

```

6.[2121] 多项式求和 1

观察下面的式子，并找出其中的规律，计算前  $n$  项和。

$1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + \dots$

输入

每行输入 1 个整数  $n$  ( $0 \leq n \leq 100$ )，当  $n=0$  时，表示输入结束，不做运算。

输出

Output

每个测试实例的输出占一行，结果保留 3 位小数。

样例输入

1

2

3

0

样例输出

1.000

0.500

0.833

```
#include<iostream>
using namespace std;
int main()
{
    int n,f;
    double sum;
    while(cin>>n){
        __1____//if(n==0)break;
    __2____//sum=0;
        f=1;
        for(int i=1;i<=n;i++){
            __3____// sum=sum+1.0*f/i;

            __4____// f=-f;

        }
        printf("%.3f\n",sum);
    }
    return 0;
}
```

## 7. [NOIP2005 普及组]判断质数

### 【问题描述】

给出一个正整数，判断这个数是否是质数。

**【输入格式】**

一个正整数  $n(1 \leq n \leq 10000)$ 。

**【输出格式】**

如果  $n$  是质数，输出“YES”；否则，输出“NO”。

```
#include <iostream>
#include <cstdio>
using namespace std;
int main() {
    int ____①____;
    scanf( "%d" , &n);
    if(n==2) puts(____②____);
    else if (____③____ || n % 2 == 0) puts( "NO" );
    else {
        i = 3;
        while(i * i <= n) {
            if(____④____) {
                puts( "NO" ); return 0;
            }
            i = i + 2;
        }
        puts( "YES" );
    }
    return 0;
}
```

## 11 逻辑推理

内容:

1. 掌握逻辑推理问题的解决方法。
2. 理解枚举思想解决问题。

### 11.1 知识剖析

逻辑推理问题的常规解决方法是，假设法。假设法推理的基本方法是：先对所给定的诸多条件中的某一个条件假设它是正确的，然后结合其他条件进行合理的推理及判断，如果推理导致矛盾，说明原假设不正确，需要重新提出一个假设，再进行合情的推理，……，直到得出的结论与提供的假设及所有的条件没有矛盾发生。如此逐一检查所有的条件，直到全部问题解决为止。在计算机世界里，常常用列表枚举法与假设结合使用。

#### 1 [7143] 破案

某地发生了一件谋杀案，警察通过排查确定杀人凶手必为 4 个嫌疑犯中的一个。以下为 4 个人的供词：

- A 说：不是我。
- B 说：是 C。
- C 说：是 D。
- D 说：C 在胡说。

已知 3 个人说了真话，1 个人说了假话，根据这个信息写一个程序来确定到底谁是凶手。

分析：

#### 2. 怎么表示 A,B,C,D?

killer=1 表示 A, killer=2, 表示 B  
killer=3 表示 C, killer=4, 表示 D

#### 2. 这么表示供词?

A 说：不是我。      killer!=1  
B 说：是 C。        killer==3  
C 说：是 D。        killer==4  
D 说：C 在胡说。    killer!=4

3. 已知 3 个人说了真话，1 个人说了假话

```
sum=(killer!=1)+(killer==3)+(killer==4)+(killer!=4);
```

```
if(sum==3)
```

```
{  
    cout<<killer;  
}
```

4. 根据这个信息写一个程序来确定到底谁是凶手。

```

#include<iostream>
using namespace std;
int main()
{
    int killer;
    int i=1,sum=0;
    while(i<=4){
        killer=i;
        sum=(killer!=1)+(killer==3)+(killer==4)+(killer!=4);
        if(sum==3)
            cout<<char(64+killer);
        i++;
    }
    return 0;
}

```

### 随堂练习 1

明明、亮亮、强强三人在社区运动场上踢足球，不小心将王老师家的玻璃窗打碎了。当王老师问他们是谁打碎了玻璃窗时，明明说：“是亮亮打的。”亮亮说：“不是我打的。”强强也说：“不是我打的。”经调查知，他们三人中只有一个人讲了实话。请问到底是谁打碎了玻璃窗

#### 3. 写出表达式

明明说：“是亮亮打的.”\_\_\_\_\_

亮亮说：“不是我打的.”\_\_\_\_\_

强强也说：“不是我打的.”\_\_\_\_\_

### 2 [2221] 推算车牌号码

一天，小明和他的爸爸妈妈在街头散步，忽然发现一辆汽车肇事逃逸。可惜他们中没有一个人记住这辆汽车的 4 位数牌号。但是，爸爸记得牌号的前 2 位数是相同的，妈妈记得牌号的 后 2 位数是相同的，而小明记得这 4 位数恰好是一个 2 位数的平方数。凭着这些线索，你能准确地确定汽车牌号吗？

输出 aabb(车牌号码)

**思考：**谈谈与鸡兔同笼问题的相同点？

```

#include<iostream>
using namespace std;

int main()
{

```

```

int a,b,c;
for(a=1;a<=9;a++)
{
    for(b=0;b<=9;b++)
    {
        for(c=10;c<=99;c++)
            if(a*1100+b*11==c*c)cout<<a*1100+b*11;
    }
}
return 0;
}

```

## 11.2 案例解析

### 3 [7627] 围棋赛

A、B、C、D、E 五人参加围棋赛，四位观战者预测了结果。

甲说：“E 第 3，A 第 4。”

乙说：“A 第 3，B 第 1。”

丙说：“B 第 4，E 第 2。”

丁说：“D 第 1，C 第 3。”

实际结果是每人只猜对了一个。参赛五人没有并列名次。输出 a,b,c,d,e 的最终名次。

**思考：采用枚举的思想解决问题。**

5. 每个人的枚举范围如何表示？

6. 写出表达式

- 甲说：“E 第 3，A 第 4。” \_\_\_\_\_
- 乙说：“A 第 3，B 第 1。” \_\_\_\_\_
- 丙说：“B 第 4，E 第 2。” \_\_\_\_\_
- 丁说：“D 第 1，C 第 3。” \_\_\_\_\_

7. 每人只猜对了一个

---

8. 甲乙丙丁的值不能相等，如何表达？

---

```

#include<bits/stdc++.h>
using namespace std;
int main () {
    int a,b,c,d,e;
    int jia,yi,bin,ding;
    for(a=1;a<=5;a++)//每个人 5 中排名
    for(b=1;b<=5;b++)

```

```

for(c=1;c<=5;c++)
for(d=1;d<=5;d++)
for(e=1;e<=5;e++)
{
  1 _____//甲乙丙丁的四句话
  2 _____
  3 _____
  4 _____
    if(5 _____){//一句话对
      int f1=(a!=b)&&(a!=c)&&(a!=d)&&(a!=e);//不重复
      int f2=(b!=c)&&(b!=d)&&(b!=e);
      int f3=(c!=d)&&(c!=e);
      int f4=(d!=e);
      //a,b,c,d,e 互不相等
      if( f1&&f2&&f3&&f4)
        cout<<a<<" "<<b<<" "<<c<<" "<<d<<" "<<e<<" "<<endl;
    }
  }
return 0;
}

```

#### 4[7628] 数学竞赛

小张、小王、小李、小赵同时参加一次数学竞赛，赛后，  
 小张说：“小李得第  $x$  名，我得第三名。”  
 小王说：“我得第一名，小赵得第四名。”  
 小李说：“小赵得第  $y$  名，我得第三名。”  
 小赵没有说话。成绩揭晓时，发现他们每个人的话都只说对了一半。  
 问，他们四个人的名次到底是怎样的。

输入  $x,y$

输出小张、小王、小李、小赵的名次。

样例输入 1 2

样例输出 3 1 4 2

**思考：**

写出表达式

小张说：“小李得第  $x$  名，我得第三名。” \_\_\_\_\_

小王说：“我得第一名，小赵得第四名。” \_\_\_\_\_

小李说：“小赵得第  $y$  名，我得第三名。” \_\_\_\_\_

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main () {
```

```
    int a,b,c, d;
```

```
    int jia,yi,bin,ding;
```

```

for(a=_____)//每个人 5 中排名
for(b=_____ )
for(c=_____ )
for(d=_____ )
{
    jia=_____//甲乙丙丁的四句话
    yi=_____
    bin=_____
    if(_____){//
        int f1=(a!=b)&&(a!=c)&&(a!=d);//不重复
        int f2=(b!=c)&&(b!=d);
        int f3=(c!=d);
        //a,b,c,d 互不相等
        if(f1&&f2&&f3)
            cout<<a<<" "<<b<<" "<<c<<" "<<d<<endl;
    }
}
return 0;
}

```

## 5 [7629] 苹果数量

刘红、陈明、李小明三人各有一些苹果。

刘红说：“我有 a 个苹果，比陈明少 2 个，比李小明多 1 个。”

陈明说：“我的苹果数不是最少的，李小明和我的苹果数差 3 个，李小明有 b 个苹果。”

李小明说：“我比刘红苹果少，刘红有 c 个苹果，陈明比刘红多 3 个苹果。”

他们每人说的三句话中，都有一句是错话。请问：他们各有多少苹果

输入 a b c

输出 他们各有多少苹果

样例输入 22 25 23

样例输出 22 25 21


```

#include<bits/stdc++.h>
using namespace std;
int main () {
    int a, b, c;// 刘红、陈明、李小明
    int jia,yi,bin,ding;
    int x,y,z;
    cin>>x>>y>>z;
    for(a=0;a<=100;a++)//每个人 5 中排名
    for(b=0;b<=100;b++)
    for(c=0;c<=100;c++)

```



```

{
    
}
return 0;
}

```

## 6 [7741]地理课

地理课上老师挂出一张没有注明省份的中国地图.其中有 5 个省份分别编上了数字 1~5 号, 请同学们写出每个编号是哪一省. (用数组实现比较好)

A 答: 2 号是陕西, 5 号是甘肃;

B 答: 2 号是湖北, 4 号是山东;

C 答: 1 号是山东, 5 号是吉林;

D 答: 3 号是湖北, 4 号是吉林;

E 答: 2 号是甘肃, 3 号是陕西.

这 5 名同学每人都只答对了一个省, 并且每个编号只有一个人答对.问从 1 号到 5 号各是哪个省

分析: 先假设 A 说的第 1 句话正确, 即“2 号是陕西”, 由 E 的回答知“2 号是甘肃”不正确, 从而只能“3 号是陕西”正确, 这与假设矛盾! 所以只能推定 A 说的第二句话“5 号是甘肃”正确, 下面只需利用条件“5 名同学只答对了一个省及每个编号只由 1 个人答对”, 推出正确的结果 1 号是山东,2 号是湖北,3 号是陕西,4 号是吉林,5 号是甘肃。

```

#include<bits/stdc++.h>
using namespace std;

int main(){//2020 哥德巴赫猜想
    //陕西 1 甘肃 2 湖北 3 山东 4 吉林 5
    int a1=1,a2,a3, a4,a5;
    int f1,f2,f3,f4,f5;
    for(a1=1;a1<=5;a1++)
        for(a2=1;a2<=5;a2++)
            for(a3=1;a3<=5;a3++)
                for(a4=1;a4<=5;a4++)
                    for(a5=1;a5<=5;a5++){
                        f1=(a2==1)+(a5==2);

```

```

        f2=(a2==3)+(a4==4);
        f3=(a1==4)+(a5==5);
        f4=(a3==3)+(a4==5);
        f5=(a2==2)+(a3==1);
        if(f1&&f2&&f3&&f4&&f5){
            if(a1~a5 都不相等)
                cout<<a1<<" "<<a2<<" "<<a3<<" "<<a4<<" "<<a5<<endl;
        }
    }
}

return 0;
}

```

思考：判断  $a1 \sim a5$  都不相等不相等如何表达？

## 7 [3177] 谁参赛了

A、B、C、D、E、F、G、H、I、J 共 10 名学生有可能参加本次计算机竞赛，也可能不参加。因为某种原因，他们是否参赛受到下列条件的约束：

1. 如果 A 参加，B 也参加；
2. 如果 C 不参加，D 也不参加；
3. A 和 C 中只能有一个人参加；
4. B 和 D 中有且仅有一个人参加；
5. D、E、F、G、H 中至少有 2 人参加；
6. C 和 G 或者都参加，或者都不参加；
7. C、E、G、I 中至多只能 2 人参加
8. 如果 E 参加，那么 F 和 G 也都参加。
9. 如果 F 参加，G、H 就不能参加
10. 如果 I、J 都不参加，H 必须参加

请编程根据这些条件判断这 10 名同学中参赛者名单。如果有多种可能，则输出所有的可能情况。每种情况占一行。参赛同学按字母升序排列，用空格分隔。

比如：

C D G J

就是一种可能的情况。

输入

思考：

1. 每个人的枚举范围如何表示？
2. 如何表达每个人的判断？

```

#include<bits/stdc++.h>
using namespace std;

```

```

#include<stdio.h>
int main () {

```

```

int a, b, c, d, e, f, g, h, i, j;
for(a = 0; a < 2; a++) {
    for(b = 0; b < 2; b++) {
        if(a == 1&&b == 0) continue;//1
        for(c = 0; c < 2; c++) {
            if(a+c == 2) continue;//3
            for(d = 0; d < 2; d++) {
                if(c == 0&&d == 1) continue;//2
                if(b+d != 1) continue;//4
                for(e = 0; e < 2; e++) {
                    for(f = 0; f < 2; f++) {
                        for(g = 0; g < 2; g++) {
                            if(c+g == 1) continue;//6
                            if(e == 1&&f+g != 2) continue;//8
                            for(h = 0; h < 2; h++) {
                                if(d+e+f+g+h < 2) continue;//5
                                if(f == 1&&g+h != 0) continue;//9
                                for(i = 0; i < 2; i++) {
                                    if(c+e+g+i > 2) continue;//7
                                    for(j = 0; j < 2; j++) {
  if(i+j == 0&&h == 0) continue;//10
  if(a == 1) printf("A ");
  if(b == 1) printf("B ");
  if(c == 1) printf("C ");
  if(d == 1) printf("D ");
  if(e == 1) printf("E ");
  if(f == 1) printf("F ");
  if(g == 1) printf("G ");
  if(h == 1) printf("H ");
  if(i == 1) printf("I ");
  if(j == 1) printf("J ");
  printf("\n");
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
return 0;
}

```

### 11.3 思维拓展：数组

在上面的题目中，碰到了多次要求判断几个数两两不相等的情况，如

```
int f1=(a!=b)&&(a!=c)&&(a!=d);//不重复
int f2=(b!=c)&&(b!=d);
int f3=(c!=d);
//a,b,c,d 互不相等
if(f1&&f2&&f3)
    cout<<a<<" "<<b<<" "<<c<<" "<<d<<endl;
```

这里变量不多，可以一一定义，如果有 5000 个呢？显然这个方法将没法处理。这里我们引入数组的概念。

我们需要把一大批具有相同性质的数据组合成一个新类型的变量，可以用简单的程序（比如循环 50 次）对这个新变量的各个分量进行相同的处理，每个分量仍然保留单个变量的所有性质。像数学中使用下标变量  $a_i$  形式表示这 50 个数，则问题就容易实现。在 C++ 语言中，具有下标性质的数据类型是数组。

我们把一大批具有相同性质的数据组合成一个新类型的变量，可以用简单的程序（比如循环 50 次）对这个新变量的各个分量进行相同的处理，每个分量仍然保留单个变量的所有性质。像数学中使用下标变量  $a_i$  形式表示这 50 个数，则问题就容易实现。在 C++ 语言中，具有下标性质的数据类型是数组。

例如，读入 50 个学生的成绩，只需写如下语句即可：

```
int a[51];
for (int i=1;i<=50;++i)
    cin>>a[i];
```

例：输入 500 个学生的某门课程的成绩，打印出低于平均分的学生序号与成绩。

```
int a[501],tot = 0;
for (int i=1;i<=500;++i)
{
    cin>>a[i];
    tot+=a[i];
}
ave= tot/50; //计算平均分
for (int i=1;i<=500;++i)
    if (a[i]<ave) cout<<"No. "<<i<<" "<<a[i];
```

定义一维数组的格式如下：

**类型标识符 数组名 [ 常量表达式 ]**;

在定义数组时必须指定数组的类型，与普通变量的定义类似，一个数组在内存中占一片连续的存储单元。其中，类型标识符可以是任何基本数据类型，相同类型的数组可以一起定义。数组名必须是合法的标识符。常量表达式的值即为数组元素的个数。

思考：怎么表示彼此不相等的简单表示法

```
int a[10000]={0};
int flag=1;
for(i=1;i<=50;i++) {
    a[a1]++;
}
for(i=1;i<=50;i++)
if(a[a1]>=2){
    flag=0;
}
```

### 课堂作业列表

|               |               |
|---------------|---------------|
| [7143] 破案     | [2221] 推算车牌号码 |
| [7627] 围棋赛    | [7628] 数学竞赛   |
| [7629] 苹果数量   | [7741] 地理课    |
| [3177] 谁参赛了   | [2663] 陶陶摘苹果  |
| [3317] 求平均值 2 | [2667] 校门外的树  |

### 课前练习

|                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1 阅读程序</p> <pre>#include &lt;iostream&gt; using namespace std; int main() {     int sum=0,maxn,i;     cin&gt;&gt;maxn;     for(i=1; i&lt;=maxn; i++)     {         if(i%2!=0)             sum=sum+i;     }     cout&lt;&lt;sum&lt;&lt;endl;     return 0; }</pre> <p>输入： 200<br/>输出： _____</p> | <p>2 阅读程序</p> <pre>#include&lt;iostream&gt; using namespace std; int main() {     int n,temp,i,sum=0,a=2;     scanf("%d" ,&amp;n);     temp=a;     for(i=0; i&lt;n; i++) {         sum+=temp;         temp=temp*10+a;     }     printf("%d\n",sum); }</pre> <p>输入 6 5<br/>输出 _____</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 课后练习

1. 以下程序段的输出结果是( )

```
int i,j,k;
```

```

for(i=1;i<5;i=i+2)
{
    k=1;
    for(j=i;j<5;j++) k=k+j;
}
printf("s=%d\n",k);
A)s=1          B)s=8          C)s=18          D)s=5

```

2. 以下程序中的变量已正确定义

```

int i,k;
for(i=0;i<4;i++,i++)
for(k=1;k<3;k++)printf("*");

```

程序段的输出结果是( )

- A)\*\*\*\*\*      B)\*\*\*\*      C)\*\*      D)\*

3. 有以下程序:

```

main()
{
    int i,s,n=0;
    for(i=1;i<=5;i++)
    {
        s=i%3;
        while(s>0) s--,n++;
    }
    printf("%d,%d",n,s);
}

```

程序运行后的输出结果是\_\_\_\_\_。

- A)5,0      B)5,-1      C)6,0      D)6,-1

4. 【元旦摆花】

元旦快到了，红旗小学为了在校门口用红花和黄花排成一个  $29 \times 29$  的鲜花方阵，需要计算红花和黄花各需要多少盆。摆花时规定最外层是红花，而且从外到内分别是一圈红花，一圈黄花，这样每层都是红花和黄花间隔。下面的图 3 展示的是  $5 \times 5$  方阵摆花的样子。

```

红 红 红 红 红
红 黄 黄 黄 红
红 黄 红 黄 红
红 黄 黄 黄 红
红 红 红 红 红

```

```

#include <iostream>
using namespace std;
const int n = 29;
int main()
{
    int red_s, yellow_s, i, j;
    bool red;

```

```

i = n;
red_s = ①;
yellow_s = 0;
red = true;
while (i > 1)
{
    if (red == true)
    {
        red_s = red_s + (i - 1) * 4;
        red = ②;
    }
    else
    {
        yellow_s = yellow_s + (i - 1) * 4;
        red = !red;
    }
    i -= 2;
}
if (red == true)
    red_s = ③;
else
    yellow_s = yellow_s + 1;
cout << "red flower=" << red_s << ",yellow flower=" << ④ << endl;
return 0;
}

```

## 12 图形绘制 1

内容:

1. 强化对多重循环的表达。
2. 掌握多重循环循环变量的联动。
3. 掌握基础图形规则。

### 1[3229] 输入图形 1

输出如下图形

输入

一个整数  $n$  ( $n < 10$ ), 如 5

输出如下图形:

```
*
**
***
****
*****
```

分析

| 行 i 取值 | 每行星号个数数 j | 关系            |
|--------|-----------|---------------|
| i=1    | j=1       | 每行*的个数 j 等于行号 |
| i=2    | j=2       |               |
| i=3    | j=3       |               |
| i=4    | j=4       |               |

```
#include<iostream>
using namespace std;
int main(){
    int i,j,n;
    cin>>n;
    for(i=1;i<=n;i++){
        for(j=1;j<=i;j++)
        cout<<"*";
        cout<<endl;
    }
    return 0;
}
```

### 2[7397] 简单图形

输入 n



输出

\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*

|           |                                         |
|-----------|-----------------------------------------|
| 样例输入<br>5 | 样例输出<br>*<br>**<br>***<br>****<br>***** |
|-----------|-----------------------------------------|

分析

| 行 i 取值 | 每行空格数 j | 每行星号个数 k | 关系                     |
|--------|---------|----------|------------------------|
| i=1    | j=3     | k=1      | i,j _____<br>i,k _____ |
| i=2    | j=2     | k=2      |                        |
| i=3    | j=1     | k=3      |                        |
| i=4    | j=0     | k=4      |                        |

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int i,n,j;
    cin>>n;
    for(i=1;i<=n;i++){
        for(j=1;j<=n-i;j++)
            cout<<" ";
        for(j=1;j<=i;j++)
            cout<<"*";
        cout<<endl;
    }
    return 0;
}
```

### 课堂作业 1

输出如图所示的 9\*9 乘法表

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

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int i,j;
    for(1_____){
        for(2_____){
            printf(3_____);"%2d*%2d=%2d",i,j,i*
            cout<<endl;
        }
        return 0;
    }
}

```

### 3[1533] 树形图案

输入数据为 1 个正整数  $N$  ( $1 < N < 10$ )， $N$  为奇数。请根据样例输出图形。

注：上半部分为底为  $N$  的等边三角形，下半部分为长度为  $(N-1)/2$  的直线。均以 '\*' 号代替。

每行行末无空格。如输入 5 时，打印出

```

*
***
*****
*
*

```

样例输入

7

样例输出

```

*
***
*****
*****
*
*
*

```

分析：以深入为 7 时说明上半部分行和列的关系

| i 取值 | 空格数 j | 星号个数 k | 关系              |
|------|-------|--------|-----------------|
| i=1  | j=3   | k=1    | $i+j=4, k=i$    |
| i=2  | j=2   | k=3    | $i+j=4, k=2i-1$ |
| i=3  | j=1   | k=5    | $i+j=4, k=2i-1$ |
| i=4  | j=0   | k=7    | $i+j=4, k=2i-1$ |

根据上述分析，可以得到 \_\_\_\_\_  $i+j=n$ (\*的行数)， $k=2*i-1$

```

#include<iostream>
using namespace std;
int main( ){
    int n,i,j;
    cin>>n;
    for(i=1;i<=(n+1)/2;i++)//上半部分
    {
        for( 1 ) cout<<" ";//空格数
        For( 2 ) cout<<"*";
        cout<<endl;
    }
    for( 3 ){//下半部分
        for( 4 )
            cout<<" ";
        cout<<"*"<<endl;
    }
    return 0;
}

```

#### 4[7180] 沙漏图形

问题：输入 n，输出正倒 n 层星号三角形。首行顶格，星号间有一空格，效果见样例  
输入样例：

3


输出样例：

```

* * *
 * *
  *
 * *
* * *


```

```

#include <iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++){
        
        cout<<endl;
    }
}

```

```

for(int i=2;i<=n;i++){
    
    cout<<endl;
}

return 0;
}

```

## 5 [2657] 画矩形

根据参数，画出矩形。输入四个参数：前两个参数为整数，依次代表矩形的高和宽（高不少于 3 行不多于 10 行，宽不少于 5 列不多于 10 列）；第三个参数是一个字符，表示用来画图的矩形符号；第四个参数为 1 或 0，0 代表空心，1 代表实心。

输入

输入一行，包括四个参数：前两个参数为整数，依次代表矩形的高和宽（高不少于 3 行不多于 10 行，宽不少于 5 列不多于 10 列）；第三个参数是一个字符，表示用来画图的矩形符号；第四个参数为 1 或 0，0 代表空心，1 代表实心。

输出画出的图形。

样例输入

7 7 @ 0

样例输出

```

@@@@@@@@
@          @
@          @
@          @
@          @
@          @
@@@@@@@@

```

分析：规律是什么，如何描述？

方法 1:

```

#include<iostream>
using namespace std;
int main(){
    int a,b,d,i,j;
    char c;
    cin>>a>>b>>c>>d;
}

```

```

if (d==0){
    for(i=1;i<=b;i++)
        cout<<c;
    cout<<endl;
    for(i=1;i<=a-2;i++){
        cout<<c;
        for(j=1;j<=b-2;j++)
            cout<<" ";
        cout<<c<<<endl;
    }
    for(i=1;i<=b;i++)
        cout<<c;
}
else {
    for(i=1;i<=b;i++){
        for(j=1;j<=a;j++)
            cout<<c;
        cout<<endl;
    }
}
return 0;
}

```

方法 2:

```

#include<iostream>
using namespace std;
int main()
{
    int c,k,flag,i,j;
    char x;
    cin>>c>>k>>x>>flag;
    for(i=1;i<=k;i++)
    {
        cout<<x;
        for(j=2;j<c;j++)
        {
            if(flag==0)
            {
                if(i==1)cout<<x;
                else if(i==k)cout<<x;
                else cout<<" ";
            }
            if(flag==1)
            {

```

```

        cout<<x;

    }
}
cout<<x<<endl;
}
}

```

## 6 [2365] 奥巴马会编程

美国总统奥巴马不仅呼吁所有人都学习编程，甚至以身作则编写代码，成为美国历史上首位编写计算机代码的总统。2014 年底，为庆祝“计算机科学教育周”正式启动，奥巴马编写了很简单的计算机代码：在屏幕上画一个正方形。现在你也跟他一起画吧！

输入

输入在一行中给出正方形边长  $N$  ( $3 \leq N \leq 20$ ) 和组成正方形边的某种字符  $C$ ，间隔一个空格。

输出

输出由给定字符  $C$  画出正方形的四边，用 '\$' 对正方形进行填充。但是注意到行间距比列间距大，所以为了让结果看上去更像正方形，我们输出的行数实际上是列数的 50%（四舍五入取整）。

样例输入

10 a

样例输出

```

aaaaaaaaaa
a$$$$$$$a
a$$$$$$$a
a$$$$$$$a
aaaaaaaaaa

```

分析：规律是什么，如何描述？

---

```

#include<iostream>
using namespace std;
int main()
{
    char ch;
    int n,i,j,m;
    cin>>n>>ch;
    m=n/2;
    if(n%2) m++;
    for(i=1;i<=m;i++){
        for(j=1;j<=n;j++){
            if(i==1 || i==m) cout<<ch;
            else if(j==1 || j==n) cout<<ch;

```

```

        else cout<<"$";
    }
    cout<<endl;
}
return 0;
}

```

## 7[1330] 打印空心等腰梯形

输入一个正整数  $n$ ，输出如下所示的高和上底均为  $n$  的等腰梯形：

输入输入数据占一行，为一个正整数  $n$ ， $3 < n \leq 10$

输出对输入文件中的每个测试数据，输出高和上底均为  $n$  的等腰梯形。

等腰梯形所在最小矩形框范围内空白处需输出空格。

样例输入

8

样例输出

```

      *****
     *       *
    *       *
   *       *
  *       *
 *       *
*       *
*****

```

```

#include<stdio.h>
int main()
{
    int n,i,j;
    scanf("%d",&n);
    for(i=1;i<=3*n-2;i++)
    {
        if(i<=n-1 || i>=2*n)
            printf(" ");
        else
            printf("*");
    }
    printf("\n");
    for(i=2;i<n;i++)
    {
        for(j=1;j<=3*n-2;j++)
        {
            if(j==n-i+1 || j==2*n+i-2)
                printf("*");

```

```

        else
            printf(" ");
    }
    printf("\n");
}
for(i=1;i<=3*n-2;i++)
    printf("*");
printf("\n");
return 0;
}

```

## 课堂作业列表

- |                 |                     |
|-----------------|---------------------|
| [3229] 输入图形 1   | [7397] 简单图形         |
| [1533] 树形图案     | [7180] 沙漏图形         |
| [2657] 画矩形      | [2365] 奥巴马会编程       |
| [1330] 打印空心等腰梯形 | [2212] 由 '*' 组成的三角形 |
| [2220] 数字组成的三角形 |                     |

## 课前练习

1. 以下程序执行后的输出结果是\_\_\_\_\_

```

int main()
{ int i,k,j;
  for(i=1;i<=9;i++)
  { for (k=1;k<=10-i;k++)
    printf(" ");
    for (j=1;j<=2*i-1;j++)
      printf("%c",48+i);
    printf("\n");
  }
}

```

2 阅读程序

```

#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    int n, x, i, temp, j, count=0;
    cin >> n >> x;
    for (i = 1; i <= n; i++)
    {

```



```

temp = i;
while (temp > 0)
{
    j = temp % 10;
    temp = temp / 10;
    if (j == x)
        count++;
}
}
cout << count << endl;
return 0;
}

```

输入:

100 5

输出:

## 课后练习

1. 以下能打印下面图形的程序段的是

```

* * *
* * *
* * *

```

A) for(i=1;i<=3;i++)

```

{   do {printf(" * ");}while (j<=3) ;
    printf("\n");
}

```

A) for(i=11;i<=3;i++)

```

{   for(j=1;j<=3;j++) printf(" * ");
    Printf ("\n") ;
}

```

C) for(i=1;i<=3;i++)

```

switch(i)
{   case 1:printf(" * ");
    case 2:printf(" * ");
    case 3:printf(" * ");
}

```

D) for(i=1;i<=3;i++)

```

{   j=1;
    while(j<=3) printf(" * ");
    printf("\n");
}

```

2. 有以下程序:

```
main( )
{
    int i=1,sum=0,n;
    scanf("%d",&n);
    do
    {
        i=i+2; sum=sum+1;
    }while(i!=n);
    printf("%d",sum);
}
```

若要使程序的输出值为 8, 则应该从键盘输入的 n 的值是( )

A)19      B)13      C)15      D)17

3. main( )

```
{
    int i,j;
    for(i=1;i<=3;i++)
    {
        for(j=1;j>=1;j--) printf(" * ");
        printf("\n");
    }
}
```

程序运行后的输出结果是\_\_\_\_\_。

A) \* \* \*      B) \*      C) \*      D) \* \* \* \* \*  
\* \*            \* \*            \* \*            \* \* \* \* \*  
\*              \* \* \*            \* \* \*

4. 有以下程序:

```
int main( )
{
    int i=1,j=9,n=0;
    while(i<j)
    {
        n=n+i+j;
        if(n>=30)break;
        i++; j--;
        n=n+i;
    }
    printf("%d\n",n);
}
```

程序运行后的输出结果是( )

A)0      B)30      C)35      D)40

5. 以下程序执行后的输出结果是\_\_\_\_\_

```
#include<iostream>
using namespace std;
int main()
{
    int i,j,n=5;
```

```

for(i=1;i<=n;i++){
for(j=1;j<=2*i-1;j++)
cout<<"*";
cout<<endl;
}
return 0;
}

```

6. 根据参数，画出矩形。输入四个参数：前两个参数为整数，依次代表矩形的高和宽（高不少于 3 行不多于 10 行，宽不少于 5 列不多于 10 列）；第三个参数是一个字符，表示用来画图的矩形符号；第四个参数为 1 或 0，0 代表空心，1 代表实心。

```

#include<iostream>
using namespace std;
int main()
{
int n,m;
char c;
int s;
cin>>n>>m>>c>>s;
if(____ 1 _____)
{
for(int i=1;i<=n;i++)
{
for(int j=1;____ 2 _____)
{
if(____ 3 _____)
cout<<c;
else
cout<<" ";
}
____ 4 _____
}
}
else
{
for(int i=1;i<=n;i++)
{
for(int j=1;j<=m;j++)
{
____ 5 _____
}
cout<<endl;
}
}
return 0;}

```



## 13 图形绘制 2

### 1 [1023]叠字母

题目描述

TOM 喜欢玩一种叠字母的游戏，游戏规则如下，从键盘上输入一个自然数  $N$ （保证输入的  $N$  满足  $1 \leq n \leq 13$ , 不需判断），输出具有如下规律的图形，其中最后一行顶格输出。下面图形是  $N=5$  时的情形。

```
AZ
ABYZ
ABCXYZ
ABCDWXYZ
ABCDEWXYZ
```

输入：输入一个  $N$ ， $1 \leq N \leq 13$ 。

输出：输出具有如下规律的图形，其中最后一行顶格输出。

| 样例输入 | 样例输出  |
|------|-------|
| 1    | AZ    |
| 2    | AZ    |
| 3    | ABYZ  |
|      | AZ    |
|      | ABYZ  |
|      | ABCXY |

分析：ASCII（American Standard Code for Information Interchange，美国信息交换标准代码）是基于拉丁字母的一套电脑编码系统，主要用于显示现代英语和其他西欧语言。它是现今最通用的单字节编码系统，

小写字母 a 对应的 ASCII 码为 97 大写字母 a 对因为小写字母 65

```
#include<stdio>
char a,b;
int main()
{
    int n;
    int i,j;
    while(scanf("%d",&n)!=EOF)//cin>>n
    {
        for(i=1;i<=n;i++)
        {
            a='A';
            b='Z'-i+1;
            for(j=1;j<=n-i;j++) printf(" ");
```

```

        for(j=1;j<=i;j++) printf("%c",a++);
        for(j=1;j<=i;j++) printf("%c",b++);
        printf("\n");

    }

}
return 0;
}

```

## 2 [7181] 层奇数三角形

问题：输入  $n$ ，输出  $n$  层的奇数三角形。三角形中的数如超过 9，只取个位数字。末行顶格，其余行首及数字间用空格填充，效果见样例

输入样例：

4

输出样例：

```

    1
   1 3
  1 3 5
 1 3 5 7

```

数据规模  $1 \leq n \leq 50$

```

#include <iostream>
using namespace std;

int main()
{
    int n;
    cin >> n;
    for(int i = 1; i <= n; i++)
    {
        for(int j = 1; j <= n - i; j++)
        {
            cout << " ";
        }
        for(int j = 1; j <= i; j++)
        {
            cout << (2 * j - 1) % 10 << " ";
        }
        cout << endl;
    }
    return 0;
}

```

### 3 [1024]图形输出

根据输入的整数  $N(1 < N < 10, \text{不需判断})$  的值，输出如下图所示的图形，该图形对应的为 4。

```
*
*****
*****
*****
*****
*****
*****
*
```

输入：输入一个  $N$ ， $1 < N < 10$ 。

输出：输出如题目描述图形。

样例输入：4

样例输出

```
*
*****
*****
*****
*****
*****
*****
*
```

分析：可分成上下两个部分，每个部分先输入空格，再输出\*号；找出行变量  $i$  与 列变量  $j$  的关系。以  $n=4$  为例，分析如下表所示：

| $i$ 取值 | 空格数 $j$    | 星号个数 $k$ | 关系                                                          |
|--------|------------|----------|-------------------------------------------------------------|
| $i=1$  | $j=6(t=0)$ | $k=1$    | $i$ 和 $j$ 的关系倒过来，假设 $t=6-j$ ，则 $t=2(i-1)$ ，可推导出 $j=2*(n-i)$ |
| $i=2$  | $j=4(t=2)$ | $k=5$    |                                                             |
| $i=3$  | $j=2(t=4)$ | $k=9$    |                                                             |
| $i=4$  | $j=0(t=6)$ | $k=13$   |                                                             |

方法 1:

```
#include<bits/stdc++.h>
using namespace std;
main(){
    int n,i,j,k;
    while(cin>>n){
        for(i=1;i<=n;i++){//上半部分
            for(j=1;j<=2*(n-i);j++){//推导出空格和行的关系
                cout<<" ";
            }
            for(k=1;k<=4*i-3;k++){//推导出*和行的关系
                cout<<"*";
            }
            cout<<endl;
        }
    }
```

```

        for(i=1;i<n;i++){//下半部分
            for(j=1;j<=2*i;j++)
                cout<<" ";
            for(j=1;j<=4*(n-i)-3;j++)
                cout<<"*";
            cout<<endl;
        }
    }return 0;
}

```

方法 2:

```

#include<iostream>
#include<cmath>
using namespace std;
int main()
{
    int n;
    int i,j;
    while(cin>>n)
    {
        for(i=1;i<=n*2-1;i++)
        {
            for(j=1;j<=abs(n-i)*2;j++)
                cout<<" ";
            for(j=1;j<=1+4*(n-1)-4*abs(n-i);j++)
                cout<<"*";
            cout<<endl;
        }
    }
    return 0;
}

```

方法 3:

```

#include<iostream>
using namespace std;
int main(){
    int n,i,j,k,l;
    while(cin>>n){
        for(i=1;i<=n;i++){
            for(j=1;j<=2*(n-i);j++)cout<<" ";
            for(j=1;j<=4*i-3;j++)cout<<"*";
            cout<<endl;
        }
        for(i=n-1;i>=1;i--){//倒过来输出

```

```

    for(j=1;j<=2*(n-i);j++)cout<<" ";
    for(j=1;j<=4*i-3;j++)cout<<"*";
    cout<<endl;
}
}
return 0;
}

```

#### 4[1503] ASCII 艺术

ASCII 艺术是通过巧妙的构思、用字符来设计一些漂亮的图形。

在本题中，要求编写程序输出一棵圣诞树，树叶的层数范围在 2~9，树干用单个 0 表示。

如下图所示：

输入

输入文件中包含一组测试数据。每个测试数据占 2 行，第一行一个正整数  $n$ ， $2 \leq n \leq 9$ ，第二行表示树的高度  $m$  ( $m \leq 20$ )。

输出

对输入文件中的每个测试数据，按样例输出所示的格式输出一棵圣诞树。每个测试数据的输出之后输出一个空行。

样例输入

6

9

样例输出

```

    1
   22
  3333
 444444
55555555
6666666666
 0
 0
 0

```

```

#include<stdio.h>
int main(){
    int m,n,k=1;
    scanf("%d\n%d",&m,&n);
    for(int i=1;i<=m;i++){
        for(int j=1;j<=m-i;j++){
            printf(" ");
        }
        for(int l=1;l<=2*(i-1)+1;l++){
            printf("%d",k);

```



```

    }
    printf("\n");
    k++;
}
for(int k=1;k<=n-m;k++){
    for(int i=1;i<=m-1;i++){
        printf(" ");
    }
    printf("0");
    printf("\n");
}
}

```

## 思维拓展

### 5[2126] 打印汉字

中华文化上下五千年，博大精深。现在，让你把“田”字打印成这样的形状：  
例如， $n=1$  时，打印出下面图形，

```

*****
* * *
*****
* * *
*****

```

输入一个  $n$  ( $0 < n \leq 15$ )。

输出

输出打印出来的图形。

样例输入

3

样例输出

```

*****
* * *
* * *
* * *
*****
* * *
* * *
* * *

```

```

*****
#include <iostream>
using namespace std;
int main(){
    int n;
    while(cin>>n){
        int hx=3+2*n;
        for(int j=1;j<=hx;j++)cout<<"*";
        cout<<endl;
        for(int j=1;j<=n;j++){
            cout<<"*";
            for(int b=1;b<=n;b++)cout<<" ";
            cout<<"*";
            for(int b=1;b<=n;b++)cout<<" ";
            cout<<"*";
            cout<<endl;
        }
        for(int j=1;j<=hx;j++)cout<<"*";
        cout<<endl;
        for(int j=1;j<=n;j++){
            cout<<"*";
            for(int b=1;b<=n;b++)cout<<" ";
            cout<<"*";
            for(int b=1;b<=n;b++)cout<<" ";
            cout<<"*";
            cout<<endl;
        }
        for(int j=1;j<=hx;j++)cout<<"*";
        cout<<endl;
    }
    return 0;
}

```

## 课堂作业列表

|             |                 |
|-------------|-----------------|
| [1023]叠字母   | [7181] 层奇数三角形   |
| [1024]图形输出  | [1503] ASCII 艺术 |
| [2126] 打印汉字 |                 |

## 课前练习

1 以下程序执行后的输出结果是\_\_\_\_\_

```

int main( )
{   int i,k,sum,m;
    k=sum=0; m=1;
    for(i=1;i<=10;i++)
        if(i%2) sum+=i;
        else if(i<=5) m*=i;
        else k=k+i;
    printf("%d,%d,%d\n",sum,m,k);
}

```

2 以下程序执行后的输出结果是\_\_\_\_\_

```

#include <iostream>
using namespace std;
int main(){
    int n,i,j;
    cin>>n;
    for(i=1;i<=n;i++)cout<<"*";
    cout<<endl;
    for(i=1;i<=(n-1)/2-2;i++){
        cout<<"*";
        for(j=1;j<=n-2;j++)cout<<" ";
        cout<<"*";
        cout<<endl;
    }
    for(i=1;i<=n;i++)cout<<"*";
    cout<<endl;
    for(i=1;i<=(n+1)/2;i++){
        cout<<"*"<<endl;
    }
    return 0;
}

```

## 课后练习

1 有以下程序:

```

main( )
{   int i,j,a=0;
    for( i=1; i<=2; i++)
    {   a++;
        for( j=0; j<=3; j++)
        {   if(j%3) break;

```

```

        a++;
    }
    a=a+j;
}
printf("%d\n",a);
}

```

程序运行后的输出结果是( )

A)6    B)7    C)3    D)12

2 有以下程序:

```

main( )
{
    int i,a=1;
    for( i=1; i<=10; i++)
    {
        if(a>=10) break;
        if(a%3==1) a=a+2;
        else a=a+i;
    }
    printf("%d\n",i);
}

```

程序运行后的输出结果是( )

A)5    B)6    C)7    D)8

3 有以下程序:

```

int main( )
{
    int i,j;
    for(i=1;i<=3;i++)
    {
        for(j=i;j>=1;j--) printf(" * ");
        printf("\n");
    }
}

```

程序运行后的输出结果是( )

A) \* \* \*    B) \*    C) \*    D) \* \* \* \* \* \*

    \* \*           \* \*           \* \*           \* \* \* \* \* \*

    \*           \* \* \*           \* \* \*           \* \* \* \*

4. 求  $S_n = a + aa + aaa + \dots + aa\dots a$  之值, 其中  $a$  是一个数字。例如  $2 + 22 + 222 + 2222$  (此时  $n = 4$ ),  $n$  由键盘输入。

```

main()
{
    int a,n,count=1,Sn=0,Tn=0;
    printf("请输入 a 和 n 的值:\n");
    scanf("%d,%d",&a,&n);
}

```

```

while (count<=____(1)____)
{
    Tn=____(2)____;
    Sn=____(3)____;
    ____ (4) ____;
}
printf("a+aa+aaa+...=%d\n",Sn);
}

```

5. 以下程序执行后的输出结果是\_\_\_\_\_

```

#include<stdio.h>
#include<math.h>
int main()
{
    int n=6,i,j;
    for(i=1;i<=3*n-2;i++)
    {
        if(i<=n-1 || i>=2*n) printf(" ");
        else printf("*");
    }
    printf("\n");
    for(i=2;i<n;i++)
    {
        for(j=1;j<=3*n-2;j++)
        {
            if(j==n-i+1 || j==2*n+i-2) printf("*");
            else printf(" ");
        }
        printf("\n");
    }
    for(i=1;i<=3*n-2;i++) printf("*");
    printf("\n");
    return 0;
}

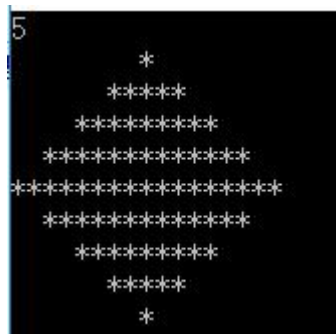
```

6. 根据输入的整数  $N(1 < N < 10, \text{不需判断})$  的值, 输出如下图所示的图形, 该图形对应的为 5。

```

#include<bits/stdc++.h>
using namespace std;
int main(){
    int n,i,j,k;
    cin>>n;
    for(i=1;i<=n;i++){
        for(j=1; ____ 1 ____ )
            cout<<" ";
        for(k=1; ____ 2 ____ )

```



```
        cout<<"*";  
        _____  
        3  
    }  
    for(i=1;i<=n-1;i++){  
        for(j=1;_____4_____)  
            cout<<" ";  
        for(k=1;_____5_____)  
            cout<<"*";  
        cout<<endl;  
    }  
    return 0;  
}
```

## 14 数学拓展

### 14.1 公约数和最小公倍数

#### 1[1433] 公约数和最小公倍数

输入两个正整数  $m$  和  $n$ ，求其最大公约数和最小公倍数。

样例输入

5 7

样例输出

1 35

提示

来源

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int a,b,x,y,t;
```

```
    cin>>a>>b;
```

```
    x=a;y=b;
```

```
    while(y!=0)
```

```
    {
```

```
        t=x%y;
```

```
        x=y;
```

```
        y=t;
```

```
    }
```

```
    cout<<x<<" "<<a*b/x;
```

```
    return 0;
```

```
}
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main ()
```

```
{
```

```
    int p,r,n,m,temp;
```

```
    cin>>n>>m;
```

```
    if(n<m)
```

```
    {
```

```
        temp=n;
```

```
        n=m;
```

```

        m=temp;
    }
    p=n*m;
    while(m!=0)
    {
        r=n%m;
        n=m;
        m=r;
    }
    cout<<n<<" ";
    cout<<p/n<<endl;
    return 0;
}

```

## 2[3044] 公约数和最小公倍数 2

输入二个正整数  $x_0$ ,  $y_0$  ( $2 \leq x_0 < 100000$ ,  $2 \leq y_0 \leq 1000000$ ), 求出满足下列条件的 P, Q 的个数。

条件:

1. P, A 是正整数;
2. 要求 P, Q 以  $x_0$  为最大公约数, 以  $y_0$  为最小公倍数。

试求: 满足条件的所有可能的两个正整数的个数。

输入的每组数据两个正整数  $x_0$  和  $y_0$  ( $2 \leq x_0 < 100000$ ,  $2 \leq y_0 \leq 1000000$ )。

对于每组输入数据, 输出满足条件的所有可能的两个正整数的个数。

下面是对样例数据的说明:

输入 3 60

此时的 P Q 分别为:

```

3    60
15   12
12   15
60   3

```

所以, 满足条件的所有可能的两个正整数的个数共 4 种。

样例输入 3 60

样例输出 4

## 14.2 环形路行程问题

在环形道路上的形成问题本质上讲就是追击问题或相遇问题。当二人(或物)同向运动时就是追击问题, 追及距离是二人初始距离及环形道路之长的倍数之和; 当二人(或物)反向运动时就是相遇问题, 相遇距离是二人从出发到相遇所行路程和。



两名运动员在沿湖的环形跑道上练习长跑。甲每分钟跑 250 米，乙每分钟跑 200 米。两人同时同地同向跑，45 分钟后甲追上了乙。如果两人同时同地反向而跑，经过多少分钟后两人相遇？

$$\begin{aligned} & (250-200) \times 45 \div (250+200) \\ &= 50 \times 45 \div 450 \\ &= 2250 \div 450 \\ &= 5 \text{ 分钟} \end{aligned}$$

### 3 [7618] 环形路行程问题

$n$  ( $n \leq 10$ ) 组运动员在沿湖的环形跑道上练习长跑。每组甲乙两名运动员，甲每分钟跑  $a$  米，乙每分钟跑  $b$  米。两人同时同地同向跑， $c$  分钟后甲追上了乙。如果两人同时同地反向而跑，经过  $t$  分钟后两人相遇。问各组运动员相遇的最短时间是多少，保留 1 位小数。

输入

第一行为  $n$ ，表示  $n$  组运动员

第二行为  $a, b, c$

第三行为  $2-n$  组运动员的速度。

输出

最短相遇时间。

样例输入

5

250 200 45

260 230

270 200

200 200

300 300

样例输出

3.8

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n,a,b,c,i=1;
```

```
    double t,mint=100000;
```

```
    int circle;
```

```
    cin>>n;
```

```
    cin>>a>>b>>c;
```

```
    circle=abs(a-b)*c;
```

```
    do{
```

```
        cin>>a>>b;
```

```
        t=1.0*circle/(a+b);
```

```
        if(t<mint)
```

```

        mint=t;
        i++;
    }while(i<n);
    printf("%.1f",mint);
    return 0;
}

```

## 14.3 分解质因数

### 4 [1329]: 质数因子分解

在数学课上，老师给同学们讲解了如何对一个正整数进行质数因子分解，如  $60=2*2*3*5$ ，其质数因子为 2、2、3、5。课后，老师给同学们布置了一道习题：求一个正整数的质数因子分解形式中，各质数因子之和（如果质数因子重复的话，则必须多次计算；注意 1 不是质数）。例如 60 的质数因子之和为 12。请你通过编程的方法帮同学们解答这道习题。

输入

每个测试数据占一行，为一个 4 位以内的正整数。

输出

对输入每个测试数据，输出该正整数的质数因子分解形式中各质数因子之和。

2008 257

1000 21

64 12

107 107

样例输入

2008

样例输出

257

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main(){
```

```
    int a,b,i,j;
```

```
    while(scanf("%d",&a)&&a!=0){
```

```
        b=0;
```

```
        for(j=1;j<=15;j++){
```

```
            for(i=2;i<=a;i++){
```

```
                if(a%i==0){
```

```
                    b+=i;
```

```
                    a=a/i;break;
```

```
            }
```

```
        }
```

```
        if(a==1){
```

```
            break;
```

```
        }
```

```

    }
    printf("%d\n",b);
}
return 0;
}

```

#### 4 [3552] 桐桐的数学难题

今天数学课上，桐桐学习了质数的知识：一个正整数如果只能被 1 和它本身整除，那么这个整数便是质数。桐桐就想：任意一个正整数是否都能分解成若干个质数相乘的形式呢？输入一个正整数  $n(2 \leq n \leq 109)$ ，把它分解成质因子相乘的形式，如果为质数则输出该数本身。如： $36=2 \times 2 \times 3 \times 3$ ； $19=19$ 。你能帮助桐桐解决这个难题吗？

输入

输入一个正整数  $n(2 \leq n \leq 109)$

输出

把它分解成质因子相乘的形式，如果为质数则输出该数本身，乘数从小到大输出。

样例输入

99

样例输出

99=3\*3\*11

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main(){
```

```
    int x,i;
```

```
    cin>>x;
```

```
    int flag=0;
```

```
    cout<<x<<"=";
```

```
    for(int i=2;i*i<=x;i++){
```

```
        while(x%i==0){//这里多次循环当前 i,直到不能整除 i 为主
```

```
            x/=i;
```

```
            if(flag==0){
```

```
                cout<<i;
```

```
                flag=1;
```

```
            }
```

```
            else{
```

```
                cout<<"*"<<i;
```

```
            }
```

```
        }
```

```
    }
```

```
    if(x!=1&&flag==0){
```

```
        cout<<x;
```

```
    }
```

```
    else if(flag==1&&x!=1){
```

```
        cout<<"*"<<x;
```

```
}  
    return 0;  
}
```

## 13.4 完数

完全数（Perfect number），又称完美数或完备数，是一些特殊的自然数。它所有的真因子（即除了自身以外的约数）的和（即因子函数），恰好等于它本身。

如果一个数恰好等于它的因子之和，则称该数为“完全数”。第一个完全数是 6，第二个完全数是 28，第三个完全数是 496，后面的完全数还有 8128、33550336 等等。截至 2018 年，相关研究者已经找到 51 个完全数。

### 课堂作业列表

2194 求累加和

2175 求阶乘

2195 根据输入，求累加和

1019 基础累加

### 课前练习

### 课后练习

## 15 综合练习

### [1] 1328: 校园健身走

2008 年是中国奥运年。为唱响“全民健身与奥运同行”这一主题，浙江财经大学于 4 月 1 日在下沙钱塘江堤举办了全体教职工“江堤健身走”活动。

活动规则如下：

1. 健身走的路线是从下沙校区南大门(起点)→校区东边钱塘江江堤→返回南大门(终点)，形成一个环线。在这一段环线上已经设置好  $N$  个景点(含起点)。
2. 每名参加“健身走”活动的教职工首先在起点领取一张活动券，并加盖起点的标识章。然后沿着路线步行，每到达一个景点，都需要加盖该景点的标识章，到达终点后，领取本次活动的纪念品。

你的任务是编写程序，计算本次“健身走”活动所需的时间。用到的信息如下：

1. 起点和终点都是南大门，包含起点在内，一共  $N$  个景点。这  $N$  个景点构成一个环线，一共  $N$  段路程，这些路程是已知的。
2. 整个活动开始时刻设为第 0 分钟。
3. 一共  $M$  批老师参加活动，假设每批老师步行速度是一致的，同时到达起点参加活动，同时抵达终点。每批老师的到达起点的时刻(分钟)、步行速度(单位距离/分钟)是已知的。
4. 每批老师步行整个路程所花的时间中，不足一分钟的以一分钟记。另外，每批老师在起点处领取活动券并加盖起点的标识章需 1 分钟，在其他景点加盖该景点的标识章需 1 分钟，到达终点后领取纪念品的时间也是 1 分钟。注意，如果存在若干批老师同时抵达一个景点，领取活动券、加盖标识章、领取纪念品等，所需的时间还是一分钟。

输入

每个测试数据的第 1 行为一个正整数  $N$ ，表示本次活动景点的个数， $N$  不超过 100。

第 2 行为  $N$  个正整数，表示  $N$  段路程的距离，这些正整数的值都不超过 100。

第 3 行为一个正整数  $M$ ，表示参加活动的老师批数， $M$  不超过 50。

接下来的  $M$  行，每行为两个正整数  $s$  和  $v$ ，分别表示这批老师到达起点的时刻，及步行速度， $s$  的值不超过 100， $v$  的值不超过 10。

输出

对每个测试数据，输出本次活动所需的时间(分钟)。

样例输入

```
6
20 30 40 50 60 70
3
0 5
9 4
6 10
```

样例输出

```

#include<iostream>
using namespace std;
int main(){
    int sum=0,n,m,t,t1,t2,t3,x,maxt=0;
    int i,v;
    cin>>n;//n 个景点
    for(i=1;i<=n;i++){//输入 n 个景点的距离，并求和
        cin>>x;
        sum=sum+x;
    }
    cin>>m;//m 批同学
    for(i=1;i<=m;i++){//输入 m 批同学
        cin>>t1>>v;//输入每批同学的出发时间和速度
        t2=sum/v;//求 t2，走路的时间
        if(sum%v!=0)t2++;//走路的时间不足一分钟算 1 分钟
        t3=n+1;//盖章的时间
        t=t1+t2+t3;//总时间
        if(t>maxt)maxt=t;//求每批的最大时间
    }
    cout<<maxt;
    return 0;
}

```

## [2]1553: 生活费 X1

快到年末啦，xiaoshua 又陷入了经济危机中，他自己都记不得这是他第几次陷入经济危机之中啦。 orz

现在 xiaoshua 想知道，在这一年中，他有几个月陷入了经济危机中，但是他很懒，懒到懒的去算这个问题。

因此，他告诉你他每个月初领了多少月钱和每个月他的花费，希望你去帮他解决这个问题。

ps: 上个月的钱可以存到下个月用,当他陷入经济危机的时候他会去问同学借钱，当每月初的领月钱的时候，优先还欠同学的钱。还有就是 xiaoshua 可能一直欠着同学钱 o(∩\_∩)o。

输入

每组数据，第一行为一个整数  $n$ ，表示 xiaoshua 每个月初能领到的月钱。（ $0 < n <= 2000$ ）

接下来是输入 12 个整数  $a_i$ （ $i$  表示第几个月份, $0 < a_i <= 2000$ ），表示 xiaoshua 在这年中 12 个月份会用掉的钱数。

数据输到文件尾；

输出

每组数据输出 1 行，表示 xiaoshua 有几个月份陷入经济危机中。

样例输入

1000

900

900  
1100  
1100  
1100  
800  
900  
900  
1100  
1100  
1100  
700

样例输出

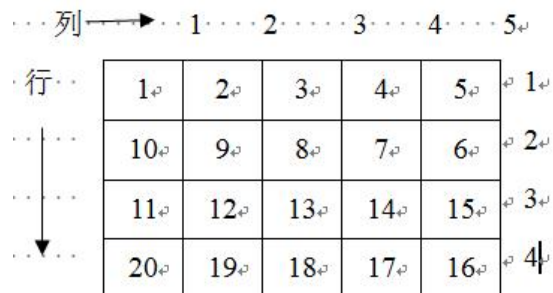
1

```
#include<iostream>
using namespace std;
int main() {
    int left = 0,monthp,use,warning = 0;
    scanf("%d",&monthp);
    for (int i = 1; i <= 12; i++) {
        left += monthp;
        scanf("%d",&use);
        left -= use;
        if (left<0) warning++;
    }
    printf("%d",warning);
    return 0;
}
```

### 3 [7314] 排座位

题目描述

为了迎接“五一”国际劳动节，笑笑所在学校决定举行庆祝活动，活动在报告厅举行，每位学生都分到了1个座位号，而报告厅的座位是按座位号蛇形排列的，学生必须对号入座。如下图是报告厅4行\*5列的座位排列情况。



现在存在的一个问题是，如何让学生根据自己的座位号很快的知道自己所坐的位置（也就是座位号所在的行、列）。

输入

第一行：包括两个数，r 和 c (1<=r,c<=100)，表示报告厅座位共有 r 行 c 列。

第二行：一个整数 n (1<=n<=r\*c)，代表有 n 个学生想知道自己的具体位置。

接下来 n 行：每一行一个整数，代表询问具体位置的座位号。

输出

输出  $n$  行，每一行两个整数，代表询问的座位号所在的行和列。

样例输入

4 5

3

4

13

19

样例输出

1 4

3 3

4 2

题目类型:模拟

思路

假设  $x$  为座位号， $t$  为行号， $h$  为列号。座位  $r$  行  $c$  列。

如果  $x\%c==0$  ( $x$  是  $c$  的整数倍)，则  $t=x/c$ 。否则 ( $x$  不是  $c$  的整数倍)  $t=x/c+1$ 。

在座位排列中，奇数行是升序排列，偶数行是降序排列。 $h=x\%c$ ，如果  $h\%c==0$ ，则  $h=c$ ，算出在奇数行中的列号，如果  $t$  是偶数，则  $h=c-h+1$ ，将升序中的列号变为降序中的列号。

```
#include<bits/stdc++.h>
#define ll long long
using namespace std;
const int N=110;
int main() {
    int r,c;
    cin>>r>>c; //输入行、列
    int n;
    cin>>n; //询问次数
    for(int i=1;i<=n;i++)
    {
        int x,t,h; //x: 输入的座位号 t: 行号 h: 列号
        cin>>x;
        if(x%c==0) t=x/c;//若x能整除c, 则t=x/c
        else t=x/c+1;//若x不能整除c, 则t=x/c+1
        cout<<t<<" ";
        h=x%c; //先假设h为升序排列（即奇数行）中的列号
        if(h==0) h=c; //若h是c的整数倍, 则在升序排列中h=c
        if(t%2==0)//如果t为偶数, 则在这一行是降序排列
        {
            h=c-h+1;//将升序的列号变成降序的列号
        }
        cout<<h<<endl;
    }
    return 0;
}
```



}

## 4[7316] 步步夺金

### 题目描述

Ivy 所在的公司想推出一款手机 APP “步步夺金”，该 APP 可以统计用户每天走路的步数，并给予相应的“金币”奖励。在用户的“金币”达到一定数量后，就可以在公司的网站上兑换一定的体育用品。

根据走路步数  $x$  奖励金币的规则如下：

- 1、每天的前 1000 步可领取 0.3 金币（若未达到 1000 步则领取数量为零），之后的每 2000 步能领取 0.1 金币。为了引导用户适量运动，每天领取的金币数量不能超过 3。
- 2、用户只有在 APP 中点击“签到”才能领取当天步数对应的金币。
- 3、为了让用户能够每天坚持使用该 APP，在用户连续三天签到后，从第四天开始，用户每天“签到”时领取的金币数可以在原金币计算方法的基础上乘以 2（当然前提是在第四天时用户也点击“签到”，并且领的金币最多不能超过 6）。但之后只要有一天中断点击“签到”，则又要重新连续签到三天，在第四天时才能继续获得金币加倍的优惠。

Ivy 的任务是，根据用户每天所走步数以及签到的情况，统计用户  $n$  天后所拥有的金币总数。请帮助 Ivy 设计这个程序。

### 输入

输入共有  $n+1$  行，第一行包含一个正整数  $n$ ，表示需要统计该用户  $n$  天后的金币总数。接下来有  $n$  ( $1 \leq n \leq 100$ ) 行，每行两个正整数， $x_i$  ( $1 \leq x_i \leq 100000$ ) 和  $f_i$  (1 或 0)，分别表示第  $i$  天时用户所走的步数以及当天该用户是否进行了签到。 $f_i$  为 0 表示用户当天未点击“签到”，为 1 表示用户当天点击“签到”。

### 输出

输出数据仅一行，表示  $n$  天后该用户所拥有的金币总数，保留一位小数。

### 样例输入

```
6
600 1
2300 1
5000 1
56000 1
80000 0
57000 1
```

### 样例输出

```
9.8
```

### 提示

第一天：用户走了 600 步，小于 1000 步，签到后得到 0 金币。

第二天：用户走了 2300 步，前 1000 步可得金币 0.3； $2300-1000=1300$ ，不足 2000 步。所以签到后可得 0.3 金币。

第三天：用户走了 5000 步，在进行签到后，可得到的金币为： $0.3+0.1*2=0.5$ 。

第四天：用户走了 56000 步，在进行签到后，应得的金币为： $0.3+0.1*27=3$ 。由于是连续第四天签到，所以实际得到  $3*2=6$  金币。

第五天：用户走了 80000 步，但未进行签到，所以得到的金币为 0。

第六天：用户走了 57000 步，进行签到后，应得的金币为： $0.3+0.1*28=3.1$

由于每天最高获金不能超过 3，所以实际得到 3 金币。

根据上述每天获得的金币，该用户 6 天后可获“金币”总数为 9.8。

题目类型:模拟

思路

根据题意计算获得的总金币数，sum 为当天获得的金币数，f 为当天的步数，x 为签到标记，t 为连续签到天数。如果 x=1，则 t+1；如果 x=0，则 t=0。如果 f>=1000 并且 x==1，那么 sum+0.3，f=f-1000。接下来计算步数每超过 2000 获得 0.1 金币，sum+=f/2000\*0.1。如果 t==4（包括当天签到），那么 sum=sum\*2。如果 sum 乘 2 之后金币数超过 6，则 sum 只能等于 6。如果 t 没有达到要求，不能乘 2，则 sum 不能超过 3，如果超过 3，则 sum 只能等于 3。ans 表示获得的总金币数，ans+=sum。

```
#include<bits/stdc++.h>
#define ll long long
using namespace std;
const int N=110;
int main() {
    int n,t=0;
    double ans=0,sum=0;
    cin>>n;//n 天
    for(int i=1;i<=n;i++)
    {
        sum=0;//当天获得金币数
        int x,f;
        cin>>f>>x;//f: 步数  x: 是否签到
        if(x==1) t++;//如果当天签到，则连续签到的天数加 1
        else t=0;//否则连续签到的天数变为 0
        if(f>=1000&&x==1)//如果 f>=1000 并且签到了，那么就获得 0.3 金币
        {
            sum+=0.3; //金币+0.3
            int h=f-1000;//h 为步数-1000，便于计算超过 2000，获得 0.1 金币
            sum+=(h/2000)*0.1;//步数减去 1000 后的步数，没超过 2000，则金币+0.1
        }
        if(t>3) //如果 t>=4,连续签到 4 天（包括今天），则金币数可以乘 2
        {
            sum*=2;
            if(sum>=6) sum=6; //当天最大金币数不能超过 6
        }
        else
        {
            if(sum>=3) sum=3;//如果没有连续签到 4 天，则当天最大金币数不能超过 3
        }
        ans+=sum;//总金币+当天获得的金币
    }
    printf("%.1f",ans);//保留一位小数
    return 0;
}
```

}

## 5[7351] 报数游戏

在一次班队活动上，班主任张老师设计了一个“报数游戏”的活动。游戏规则是这样的：每次游戏有甲、乙二位同学参加，甲按  $1-a$  的顺序循环报数，乙按  $1-b$  的顺序循环报数。两人同时开始，并以同样的速度报数，当两人都报了  $n$  个数时，统计出两人同时报相同数的次数，先算对者获胜。现在老师请你来做裁判，算出每次游戏的正确答案。

输入

共两行。第一行仅有一个整数  $n(n \leq 100)$ ，第二行有二个整数  $a、b$ ，中间用空格间隔 ( $2 \leq a、b \leq 10$ )。

输出

只有一行，有一个整数，表示两人同时报相同数的次数。

样例输入

10

2 3

样例输出

4

题目类型：模拟

思路：

该题比较简单，只要定义变量来记录每一次报数时甲乙两方的报数情况即可，相同则加

1

代码：

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int n;int sum=1;
    int a,b;
    cin>>n;
    cin>>a>>b;
    n--;
    int aa=1,bb=1;
    while(n--){
        aa=(aa+1)%(a+1);
        if(aa==0)
            aa=1;
        bb=(bb+1)%(b+1);
        if(bb==0)
            bb=1;
        if(aa==bb)
            sum++;
    }
    cout<<sum;
```

```
    return 0;
}
```

## 6[7332] 最佳时间

泽泽不但喜欢看拳击比赛,而且也喜欢下围棋和编程,所以他决定参加围棋和编程兴趣班。围棋兴趣班共有  $n$  个时间段选择,第  $i$  个时间段排在  $(L_i \sim R_i)$ 。编程兴趣班也有  $m$  个时间段选择,第  $i$  个时间段排在  $(A_i \sim B_i)$ 。他必须要选择一个围棋班和一个编程班的时间段,但他希望选的这两个班中间的休息时间越长越好。

例如,他选了这两个时间段  $(L_1 \sim R_1)$  和  $(A_1 \sim B_1)$ ,假设  $(L_1 \sim R_1)$  这节课在前面, $(A_1 \sim B_1)$  这节课在后面,那么,泽泽在中间休息的时间是  $A_1 - R_1$ 。特别说明,当两节课上课时间有冲突,泽泽在中间休息时间为  $0$ 。

泽泽想算一算他所选的两节兴趣课之间,能休息的时间最长是多少?请你帮助泽泽找一找,算一算。

输入

第一行输入一个整数  $n$ ,表示围棋兴趣班可选择的时间段。

下列  $n$  行,每行都输入两个整数  $L_i$  和  $R_i$ ,分别表示泽泽参加第  $i$  个围棋班的起止时间。

下面一行输入的一个整数  $m$ ,表示编程兴趣班可选择的时间段。

下列  $m$  行,每行都输入两个整数  $A_i$  和  $B_i$ ,分别表示泽泽参加第  $i$  个编程班的起止时间。

输出

输出一个整数,表示两个时间段之间的最长休息时间(如果所有时间段都有冲突,则输出  $0$ )。

样例输入

```
3
1 5
2 6
2 3
2
2 4
6 8
```

样例输出

```
3
```

想要有最长的休息时间,就需要令前一节课的结束时间最早、后一节课的开始时间最晚,输入时顺便求出相应的最值即可。注意如果所有时间段都有冲突,则输出  $0$ 。

思路:将围棋班的最晚上课时间-编程班的最早下课时间

将编程班的最晚上课时间-围棋班的最早下课时间,两者比较就是最长休息时间

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long
const int inf=0x3f3f3f3f3f3f3f;
ll a1,b1,a2,b2;
int main(){
```

```

int n, m;
ll min1=inf, min2=inf, max1=0, max2=0;
cin>>n;
for(int i=1;i<=n;i++){
    scanf("%lld%lld",&a1,&b1);
    if(min1>b1)min1=b1;//求最早的结束时间
    if(max1<a1)max1=a1;//求最晚的起始时间
}
cin>>m;
for(int i=1;i<=m;i++){
    scanf("%lld%lld",&a2,&b2);
    if(max2<a2)max2=a2;
    if(min2>b2)min2=b2;
}
ll ans=max(max2-min1, max1-min2);
if(ans<0)ans=0;//休息时间为负时输出0
cout<<ans;
return 0;
}

```

## 7[7354] 换座位

聪聪和同学们正在玩这样一个换座位的游戏：班上共有  $2n$  个少先队员，开始时每个少先队员坐在自己的板凳上排成一队，由聪聪开始击鼓，每次击鼓开始时，前  $n$  个同学坐到第 2、4、...、 $2n$  个板凳上，后  $n$  个同学坐到第 1、3、...、 $2n-1$  个板凳上，击鼓结束时坐错或者还没有坐到对应板凳上的同学就要接受惩罚——表演一个节目，并按规定坐好。聪聪不断的击鼓然后停顿后又击鼓...，同学们都觉得这个游戏很好玩，但是当游戏结束时，同学们傻眼了，由于每位同学的板凳都差不多，他们找不到自己的板凳了。而聪聪这时反应特别快，他说经过一定次数的换座位，每位同学一定能回到自己的板凳的。那么这个次数最少是多少呢？你会计算吗？

输入

共一行，一个正整数  $n$ 。

$1 \leq n \leq 20000$

输出

共一行，一个正整数，表示每位同学都回到自己板凳的最少换座位次数。

样例输入

10

样例输出

6

思路：该题  $N$  小于 20000，如果考虑全体成员都归为的情况，通过定义多个数组进行模拟，那么时间一定超限；不难发现，如果改队伍中只要一个成员归位，那么剩余的成员也一定是归位的，那么只要模拟其中一位成员的位置变化就可以了。

代码：

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    scanf("%d",&n);
    int sum=1;
    int flag=2;
    while(flag!=1){
        if(flag<=n){
            flag=2*flag;
        }
        else{
            flag=2*(flag-n)-1;
        }
        sum++;
    }
    printf("%d",sum);
    return 0;
}
```