

# C/C++语言入门

## 第一课 导入

### 一、 什么是 C/C++ 语言： C++语言是一种高级程序设计语言。

C++语言是 20 世纪 80 年代初期由贝尔实验室设计的一种在 C 语言基础上增加了对面向对象程序设计支持的语言，它是目前应用最为广泛的编程语言。

程序体验：

```
#include <iostream>
using namespace std;
int main()
{
    printf("hello!");
    system("pause");
    return 0;
}
```

例一：已知长方形的长、宽，求长方形的周长、面积

```
#include <iostream>
using namespace std;
int main()
{
    int a,b,L,S;
    scanf("%d%d",&a,&b);
    L=2*(a+b);
    S=a*b;
    printf("L=%d,S=%d",L,S);
    system("pause");
    return 0;
}
```

### 二、 程序设计的要素：

- 1、分析题目中的已知和求解的信息，以变量的形式定义下来。（例题中已知长和宽，求周长和面积。a,b,l,s 分别为长、宽、周长、面积）
- 2、如何求解—算法。

### 三、 C++ 程序的组成

```
#include <iostream> //标准输入输出库文件
using namespace std; //使用 std 命名空间
int main()
{
    int a,b,L,S; //定义变量
    scanf("%d%d",&a,&b); //输入
    L=2*(a+b); //处理
    S=a*b;
    printf("L=%d,S=%d",L,S); //输出
    system("pause");//查看结果
    return 0;
}
```

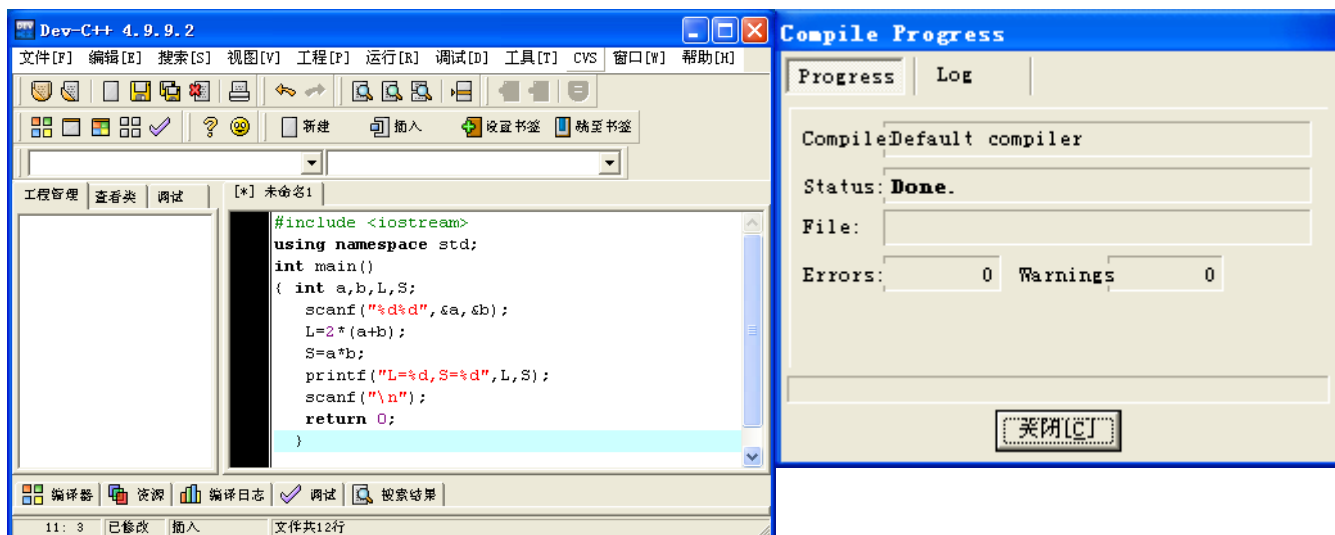
一个程序:

头文件

```
int main()
{
    变量说明部分;
    执行部分;
    system("pause");
    return 0;
}
```

执行部分主要分为: 输入、处理、输出

### 四、 C++ 程序设计的界面及主要命令



### 五、 常用操作

- 1、 新建\源代码。
- 2、 编译并运行 F9
- 3、 编译—Ctrl +F9

#### 练习

- 1、 启动 Dev C++。
- 2、 输入以上例 1 的程序
- 3、 文件保存, 编译, 运行, 看结果

书写程序时注意:

- 1、 保持缩进
- 2、 { } 要成对出现
- 3、 除了头文件, int main() 外  
每句程序后面都要加;

## 第二课 简单程序的书写

### 一、分析例题 2.1

```
例二：已知正方形的边长，求正方形的周长、面积
#include <iostream> //标准输入输出库文件
using namespace std; //使用 std 命名空间

int main()
{
    int a,L,S;
    scanf("%d",&a);
    L=4*a;
    S=a*a;
    printf("L=%d,S=%d",L,S);
    system("pause");
    return 0;
}
```

int 数据类型中的整数类型。

scanf 输入语句      =      赋值语句      printf 输出语句

### 二、变量声明部分

**格式：**类型名 变量名 1,变量名 2,...,变量名 n;

例如：int a; (单个变量说明)

float L,S; (多个相同变量同时定义,变量名之间用逗号隔开)

**变量名：**由字母、数字、下划线组成的。以字符或者下划线开头。区分大小写。不能与保留字同名。

常见的保留字：int, float,char,string, bool,unsigned,  
if ,else , do ,while, char, break  
void ,typedef,switch,register  
(c 语言中保留字、标准库函数名都只用小写字母表示)

### 三、基本数据类型:

类型	说明	字节	取值范围
[signed] char	字符型	1	-128~127
unsigned char	无符号字符型	1	0~255
[signed] short	短整型	2	-32768~32767
unsigned short	无符号短整型	2	0~65536
[signed] int	整型	4	-2147483648~2147483647
unsigned int	无符号整型	4	0~4294967295
[signed] long	长整型	4	-2147483648~2147483647
unsigned long	无符号长整型	4	0~4294967295
[signed]long long	超长整型	8	$-2^{63} \sim 2^{63}-1$
float	单精度浮点型	4	$-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$ (约 6 个有效数字)
double	双精度浮点型	8	$-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$ (约 12 个有效数字)

表中的[]表示其中的内容可以省略。如[signed] char, 简略为 char 就可以。

**注意：不同的数据类型,存储空间大小和能合适的运算不同**

#### 1、整型适用的运算操作

加	减	乘	除	余数
+	-	*	/	%

运算优先级: ( $*$  等于  $/$  等于  $\%$ ) > ( $+$  等于  $-$ )

体验: 运算符号与数据类型

```
#include <iostream> //标准输入输出库文件
using namespace std; //使用 std 命名空间
int main()
{ int a,b;
  a=5/2;b=5%2;
  printf("%d %d",a,b)
  system("pause");
  return 0;
}
```

**例题：**已知 a,b 两个整数，求这两个数的和、差、乘积、商、余数

分析：已知 a,b，求：s1(和)，s2(差)，s3(乘积)，s4(商)，y(余数)

方法：s1=a+b;s2=a-b;s3=a\*b;s4=a/b;s5=a%b;

```
#include <iostream>
using namespace std;
int main()
{ int a,b,s1,s2,s3,s4,y;
  scanf("%d%d",&a,&b);
  s1=a+b; s2=a-b; s3=a*b;
  s4=a/b; y=a%b;
  printf("%d+%d=%d\n",a,b,s1);
  printf("%d-%d=%d\n",a,b,s2);
  printf("%d*%d=%d\n",a,b,s3);
  printf("%d/%d=%d...%d\n",a,b,s4,y);
  system("pause");
  return 0;
}
```

说明：书写程序时，不一定要严格一行一个语句，可以一行多个语句。

变量名的命名与实际存储内容相关联。便于自己编程中不会混淆变量。

例题：

## 2、整型变量常用的函数

使用下列函数时，必须加入#include <cmath>

函数名	功能	函数的定义	举例
abs	求绝对值	abs(任意的数值类型)	abs(-2) 结果为 2, abs(2) 结果为 2
sqrt	求平方根	sqrt(任意的数值类型)	sqrt(4), 结果为 2; sqrt(1.5) 结果为 1.224745
int	强制类型 转换整型	int (变量名或者表达式) 一般用来把实数转换为整数	int (2.5); 结果为 2

### 3、实型合适的运算符与函数

**//double 类型精度高，在实际应用中不使用 float，使用 double 类型。**

加	减	乘	除
+	-	*	/

```
int a=5/2; //结果为 a=2
char ch=101; //int a=ch/3; 结果为 a=33;
float a=5/2; //结果为 a=2.000; 精度丢失在计算 5/2 时发生了。
double a=5.0/2;
double a=5/2.0; // 结果为 a=2.5
double a=5.0/2.0;
```

除数与被除数至少需要一个为实型，除运算才能实数除。  
如果遇到 a,b 均为整型，a/b 结果为实型，a\*1.0/b

试一试强制类型转换 double (变量), float (变量) , int (变量)

```
#include <iostream>
using namespace std;
int main()
{ int a,b;
  float c;
  double d;
  a =int(7.6); b=5;
  c = a/b; printf("%0.2f\n",c); //结果为 c=1.00
  c =float(a)/b; printf("%0.2f\n",c); //结果为 c=1.40
  c =(float)(a/b);printf("%0.2f\n",c); //结果为 c=1.00
  d= double(a)/b ;printf("%0.3lf\n",d); //结果为 d=1.400
  system("pause");
  return 0;
}
```

实数的常用函数 `#include <cmath>`

函数名	功能	实例
ceil	向上舍入 double ceil(double x)	Ceil(5.4) 结果为 5
floor	向下舍入 double floor(double x)	Floor(5.6) 结果为 6
Fabs	求浮点数的绝对值 double fabs(double x)	Fabs(-5.4) 结果为 5.4
Fmod	求 x/y 的余数 Double fmod(double x,double y)	Fmod(2,5.0) 结果为 2.0 注意不可以为 fmod(2,5)
pow	指定数的指定次幂 double pow(double x,double y)	Pow(5,3.0) 结果为 125
float	强制类型转换	float(5), 结果为 5.0
double	强制类型转换	double(6), 结果为 6.0

**练习:** 已知圆的半径, 求圆的面积和周长

**常量:** 定义: 在程序执行过程中始终保持不变的数据量。

```
#define PI 3.1415926 (注意 c++语言中区分大小写) //实数常量
#define N 100 //整数常量
#define Name "hello" //字符串常量
#define ch 'A' //字符常量
```

**常量:**

```
const double PI=4.0*atan(1.0); //实数常量
const 关键词表明它的值是不可以改变的
```

数据存储: 已知半径  $r$ , 数据类型实型, 求周长  $l$ , 面积  $s$ 。

算法:  $l=2*PI*r$   $s=PI*r*r$ ;

```
#include <iostream>
#include <cmath>
#define PI 3.1415926
using namespace std;
int main()
{ double r,l,s;
  const double PI=acos(-1);
  r=5;
  l=2*PI*r; s=PI*r*r;
  printf("l=%0.2lf,s=%0.2lf",l,s);
  system("pause");
  return 0;
}
```

4、字符类型, 实际存储的是 0..255 的整数

**定义:** `char ch1,ch2,ch3;`

**赋值:** `ch1=97;` 相当于 `ch1='a';`

`ch2='A';` 相当于 `ch2=65;`

`ch3='0';` 相当于 `ch3=48;`

**输入:** `scanf("%c%c",&ch1,&ch2);` 输入 ab

```
scanf("%c %c",&ch1,&ch2);输入 a b
```

输出: `printf("%c,%c,%c", ch1,ch2,ch3);` 输出 a,A,0

```
printf("%5c",ch1); 定义场宽为 5。
```

字符类型处理过程中注意事项:

```
#include <iostream>
using namespace std;
int main()
{ int n;
  char ch;
  scanf("%d",&n);
  scanf("%c",&ch);
  printf("%d\n",n);
  printf("%d\n",ch);
  system("pause");
  return 0;
}
```

```
输入:5
      A
实际 n=5
ch='\n'
(回车符)
ch=10
```

```
输入:5_ A
      A
实际 n=5
ch='_'
(回车符)
ch=32
```

回车符, 空格都作为字符读入

## 读入处理

情况 1: 整数 字符, 例如 5 A `scanf("%d %c",&n,&ch);`

情况 2: 整数 例如: `scanf("%d",&n);`  
字符 5 `scanf("%c",&ch);`  
A

### 练习:

- 1、输入字母序号, 输出相应的大小写字母。输入 1, 输出 A, a 输入 26, 输出 Z, z
- 2、输入一个小写字母, 将其转换为大写字母
- 3、输入字母, 输出其在字母表中的序号。

## 三、输出语句: printf 函数

语句格式: `printf("格式控制字符串", 地址列表)`

格式控制字符串:

用于指定输出格式, 由格式字符串和非格式字符串构成。



格式字符串：以%开头，后面有各种格式字符，以说明输出数据的类型、形式、长度、小数位数等。

非格式字符串：原样输出。

地址列表：各个输出项。要求格式字符串和各输出项在数量和类型上应该是一一对应的。

```
//不同的类型控制:
#include <iostream>
using namespace std;
int main()
{ int a=65,b=97;
  printf("*****\n"); //原样输出, 后换行
  printf("%d %d\n",a,b); // 65 97 %d----十进制
  printf("%d,%d\n",a,b); // 65,97
  printf("%o,%o\n",a,b); // %o-----八进制
  printf("%x,%x\n",a,b); // %x-----十六进制
  printf("%c,%c\n",a,b); // A,a %c-----字符格式
  system("pause");
  return 0;
}
```

类型格式符号	意义
d	十进制带符号整数 (整数不输出符号)
O	以八进制带符号整数 (不输出前缀 o)
X	以十六进制带符号整数 (不输出前缀 0x)
u	以十进制无符号整数
f	以小数形式输出单、双精度实数
e	以指数形式输出单、双精度实数
c	输出单个字符
s	输出字符串

“格式控制字符串” : % [输出宽度][.精度] d

```
//体验宽度和精度的控制
#include <iostream>
using namespace std;
int main()
{ int a=123; double b=123.456;
  printf("-----分割线-----\n"); //原样输出, 后换行
  printf("a=%d,%5d\n",a,a);
  printf("b=%lf,%8.5lf,%0.11f\n",b,b,b);
}
```

```
system("pause");
return 0;
}
```

//体验 long long 类型的输入输出

```
#include <iostream>
using namespace std;
int main()
{ long long a;
  scanf("%I64d",&a); //读入 123456789123456789
  printf("%I64d\n",a);
  system("pause");
  return 0;
}
```

### 三、输入语句：scanf 函数 (格式输入函数)

语句格式： scanf("格式控制字符串",地址列表)

```
#include <iostream>
using namespace std;
int main()
{
  int a1,b1,a2,b2;
  scanf("%d%d",&a1,&b1); printf("a1=%d,b1=%d\n",a1,b1);
  scanf("%d,%d",&a2,&b2);printf("a2=%d,b2=%d\n",a2,b2);
  system("pause");
  return 0;
}
```

从键盘上输入： 1 2  
                  3,4

输出： a1=1,b1=2  
       a2=3,b2=4

从键盘上读入： 1  
                  2  
                  3,

输出： a1=1,b1=2  
       a2=3,b2=4

注意:

整数与实数一般用空格或回车来区分不同变量的数据。

如何输入格式中有，输入数据一定要有，否则数据出错。

scanf 中常见的错误:

错误	正确	
<code>scanf("%d", a);</code>	<code>scanf("%d", &amp;a);</code>	没有&
<code>scanf("%5.2lf", b);</code>	<code>scanf("%f", &amp;b);</code>	不能有精度控制

//体验输入不同类型的数据

```
#include <iostream>
using namespace std;
int main()
{
    int a,b;
    double c,d;
    char ch1,ch2;
    scanf("%d%d",&a,&b);
    scanf("%lf%lf",&c,&d);
    scanf("\n%c%c",&ch1,&ch2);
    printf("a=%d,b=%d\n",a,b);
    printf("c=%.2lf,d=%0.2lf\n",c,d);
    printf("ch1=%c,ch2=%c\n",ch1,ch2);
    system("pause");
    return 0;
}
```

输入数据: 1 2

3.4 5.6

ab

输出: a=1,b=2

c=3.40,d=5.60

ch1=a,ch2=b

## 四、赋值语句 语句格式: 变量=表达式

### 1、 相同类型的赋值

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

```
a=1; b=2; c=3 ; a=b+c; 可以写成 a=(b=2)+(c=3); a=(b=2,c=3,b+c);
```

```
a=1;b=1; c=1; 可以写成 a=b=c=1;
```

```
double x,y;
```

```
x=1.1; y=x;
```

## 2、 不同类型的赋值

右边的表达式的值赋值给左边的变量时，如果两边的类型不一致时，自动进行类型转换，将右边的类型转换成左边的类型。

实型赋值给整型，舍去小数部分。

整数赋值给实型，数值不变。

字符型赋值给整型，其 ASCII 码的值放入整型的低八位，高八位赋值为 0。

整型赋值给字符型，只把低八位赋值给字符型。

## 3、复合赋值表达式

双目运算符的使用 += , -=, \*=, /=, %=, <<=, >>=, &=, ^=, |=

复合赋值表达式:

变量+双目运算符+表达式 等价于 变量=变量+运算符+表达式

a+=5 等价于 a=a+5

x\*=y+7 等价于 x=x\*(y+7)

r%=p 等价于 r=r%p

复合赋值表达式能提高编译的效率并产生质量较高的目标代码。

**例:**输入 a,b 的值，将其值交换后，输出

说明：程序 1 中，temp 为中间暂存变量；理解程序 2 为什么可以实现 a , b 值的交换？

```
int main()
{
    int a,b,t;
    scanf("%d%d",&a,&b);
    t=a;a=b;b=t;
    printf("a=%d,b=%d\n",a,b);
    system("pause");
    return 0;
}
```

```
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    a=a+b;b=a-b;a=a-b;
    printf("a=%d,b=%d\n",a,b);
    system("pause");
    return 0;
}
```

例题 2-1 交换变量。输入两个整数 a 和 b，交换二者的值，然后输出。

例题 2-2 三位数反转。输入一个三位数后，分离出百位、十位和个位，反转后输出

例题 2-3 平均数 (average) 输入 3 个整数，输出他们的平均数，保留 3 位小数。

---

例题 2-4 温度 (temperature) 输入华氏温度  $f$ , 输出对应的摄氏温度  $c$ , 保留 3 为小数。

$$c = 5 * (f - 32) / 9$$

例题 2-5 距离 (distance) 输入 4 个浮点数  $x_1, y_1, x_2, y_2$ , 求平面坐标系中点  $(x_1, y_1)$  到点  $(x_2, y_2)$  的距离

## 第三课 选择结构语句 (if、switch)

### 一、布尔类型

定义: `bool f1, f2;`

赋值: `f1=true;` 等价于 `f1=1;`

`f2=false;` 等价于 `f2=0;`

输出: `printf("%d,%d", f1, f2);` 输出为 1,0

### 二、关系运算符

作用	操作符
大于	>
小于	<
大于等于	>=
小于等于	<=
等于	==
不等于	!=

### 三、逻辑运算符

作用	操作符	优先级
非	!	高
逻辑与	&&	中
逻辑或		低

P	Q	!P	P && Q	P    Q
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

由关系运算符和逻辑运算符连接的表达式，结果为布尔类型。

### 四、优先级

算术运算符 > 关系运算符 > 逻辑运算符

常用的表达式	意义
<code>1&gt;0</code> <code>1!=0</code> <code>1==0</code> <code>1+2&gt;5</code>	结果为 true 或者 1 结果为 true 或者 1 结果为 false 或者 0 结果为 false 或者 0
<code>a&gt;0</code> <code>a==0</code> <code>a!=0</code> <code>a==b</code> <code>a!=b</code>	变量 a 是否大于 0 变量 a 是否等于 0 变量 a 是否不等于 0 变量 a 与 b 是否相等 变量 a 与 b 是否不相等
<code>a&gt;=0 &amp;&amp; a&lt;=10</code>	变量 a 的值是否在区间 [0, 10]
<code>a%4==0</code>	变量 a 是否为 4 的倍数
<code>a+b&gt;c &amp;&amp; b+c&gt;a &amp;&amp; a+c&gt;b</code> 等价于 <code>(a+b&gt;c) &amp;&amp; (b+c&gt;a) &amp;&amp; (a+c&gt;b)</code>	三角形的任意两边之和大于第三边

## 五、IF 语句

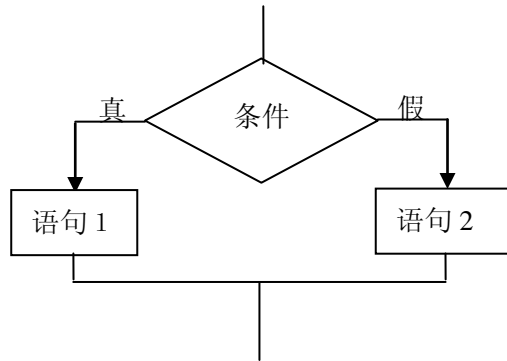
**【作用】：** 根据条件来选择执行的语句。

**【格式】：**

形式 1: if (条件) 语句;

形式 2:

```
if (条件)
    语句 1;
else 语句 2;
```



**【注意】**

- 1、条件，是一个表达式，() 不能省略。
- 2、注意加;的位置
- 3、书写 if 语句的关键在于给出条件表达式，主要使用关系表达式或者逻辑表达式来表示某一条件。

**【例题】**

**例 3.1:** 输入不相等的两个整数 a,b, 输出较大的数。

分析: 根据判断两个数的大小关系, 给出条件表达式  $a > b$ , 条件成立输出 a, 条件不成立, 输出 b; 程序中引进一个新的变量 max 来存储最大值。

```
#include <iostream>
using namespace std;
int main()
{ int a,b;
  cin>>a>>b;
  if (a>b) cout<<"max="<<a<<endl;
  else cout<<b<<endl;
  system("pause");
  return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{ int a,b,max;
  cin>>a>>b;
  if (a>b) max=a; else max=b;
  cout<<"max="<<max<<endl;
  system("pause");
  return 0;
}
```

---

**例 3.2:** 输入一个整数  $a$ , 判断是否在开区间  $(0, 8)$  内, 是, 输出 "Yes", 否则输出 "No".  
分析:  $a$  是否在开区间  $(0, 8)$  内, 也就是  $0 < a < 8$ ; 如果是闭区间  $[0, 8]$ , 那就是  $0 \leq a \leq 8$ .

表达式书写: 错误写法:  $0 < a < 8$  正确的写法:  $a > 0 \ \&\& \ a < 8$

**知识点:** 当判断由多个条件构成时, 条件之间根据逻辑运算符 **!, &&, ||** 来连接。

```
#include <iostream>
using namespace std;
int main()
{   int a;
    cin >> a;
    if (a > 0 && a < 8) cout << "Yes" << endl;
    else cout << "No" << endl;
        system("pause");
    return 0;
}
```

**例 3.3:** 输入年份, 判断是否为闰年。闰年的规则: 年份能够被 4 整除, 但不能被 100 整除, 或能被 400 整除。

```
#include <iostream>
using namespace std;
int main()
{   int year;
    cin >> year;
    if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
        cout << year << "是闰年" << endl;
    else cout << year << "不是闰年" << endl;
    system("pause");
    return 0;
}
```

**例 3.4:** 输入一个整数  $x$ , 判断该数的奇偶性, 如果是奇数, 输出 "奇数", 否则输出 "偶数"

```
#include <iostream>
using namespace std;
int main()
{   int x;
    cin >> x;
    if (x % 2 != 0) cout << x << "是奇数" << endl;
    else cout << x << "是偶数" << endl;
    system("pause");
    return 0;
}
```

请你用另外一种方法来完成该程序。

**例 3.5:** 输入两个整数  $max$ 、 $min$ , 将较大的数存在  $max$ , 较小的数放在  $min$  里。

分析: 如果  $max < min$ , 就交换数据。

**知识点:** If 语句中, 如果 <语句 1> 或者 <语句 2>, 执行的内容无法用一个语句实现, 需要用多个语句完成时, 这时候必须使用**复合语句**。

复合语句格式: { 语句 1;



```

        语句 2;
        .....;
    }

#include <iostream>
using namespace std;
int main()
{ int min,max,t;
  cin>>min>>max;
  if (min>max)
  { t=min; min=max; max=t; }
  cout<<"max="<<max<<" ,min="<<min<<endl;
  system("pause");
  return 0;
}

```

**例 3.6:** 输入三个整数 x、y、z，求其中最大值(最小值)。

分析: 给出框图

**知识点:** if 语句可以多层嵌套。

```

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

```

if 语句的嵌套  
1、if 语句格式对齐  
2、合理的缩进

**知识点:** 随机数

```

#include <iostream>
#include <ctime>
using namespace std;
int main()
{ srand((unsigned)time(NULL)); //用当前时间作为随机种子
  printf("%d\n",rand()%10); //产生 0..9 的随机数
  system("pause");
  return 0;
}

```

---

**例 3.7:** 求一元二次方程  $ax^2+bx+c=0$  的根. 输入  $a, b, c$ ; 输出解.

```
#include <iostream>
#include <cmath>
using namespace std;
double a,b,c,x1,x2,delt;
int main()
{
    cin>>a>>b>>c;
    if (a==0)
    { printf("不是一元二次方程\n");
      system("pause\n");
      return 0;// 中断程序,后面的语句不运行.
    }

    delt=b*b-4*a*c;
    if (delt>=0)
    {
        delt=sqrt(delt);
        x1=(-b+delt)/(2*a);x2=(-b-delt)/(2*a);
        printf("x1=%.4lf x2=%.4lf\n",x1,x2);
    }
    else
    {
        printf("无解\n");
    }

    system("pause\n");
    return 0;
}
```

### 练习:

- 1) 输入一个整数判断该数是否为完全平方数
- 2) 输入一个三位数, 判断这个数中是否包含一个数字 0, 如果有输出 "yes"
- 3) 输入一个数, 判断该数是正数、负数还是零, 为正数输出 "正数", 为负数输出 "负数", 为零输出 "零"
- 4) 输入三个整数, 求最小值、最大值
- 5) 输入三个整数, 将该三个数从小到大输出。
- 6) 从键盘输入三个正整数  $a, b, c$  分别表示一个三角形的三条边, 编程确定是否能构成三角形, 若能, 则输出三角形的类型 (任意三角形, 等腰三角形, 等边三角形, 直角三角形, 等腰直角三角形, 无法构成三角形)
- 7) 鸡兔同笼问题  
已知鸡和兔的总数量为  $n$ , 总腿数为  $m$ 。输入  $n$  和  $m$ , 依次输出鸡的数目和兔的数目。  
如果无解, 则输出 "No answer"

---

## 六、分情况 switch

**【作用】:** 多分支选择语句, 根据表达式的值分情况。

**【格式】:**

```
Switch (表达式)
{ case 值 1: 语句组 1; break;
  case 值 2: 语句组 2; break;
  .....
  case 值 n: 语句组 n; break;
  default: 语句组 n+1; break;
}
```

**【注意】:**

1. 注意符号 {}, (), ; : 的使用
3. 表达式的值必须为有序类型—整型、字符型
4. 语句组 n, 可以不加 {}
5. default 子句可以省略。case 和 default 顺序可改变。
6. 书写程序时, 注意缩进和对齐。

**例 3.7:**

输入两个实数 a、b 和一个运算符 opr ( '+' , '-' , '\*' , '/' ) , 输出其运算结果。

```
#include <iostream>
using namespace std;
int main()
{ char ch;
  double a,b;
  scanf("%lf%c%lf",&a,&ch,&b);
  switch (ch)
  {case '+':cout<<a<<ch<<b<< '='<<a+b<<endl;break;
    case '-':cout<<a<<ch<<b<< '='<<a-b<<endl;break;
    case '*':cout<<a<<ch<<b<< '='<<a*b<<endl;break;
    case '/':cout<<a<<ch<<b<< '='<<a/b<<endl;break;
    default:cout<<"输入错误"<<endl;
  }
  system("pause");
  return 0;
}
```

**例 3.8:** 从键盘上读入年和月，输出该月有多少天。

```
#include <iostream>
using namespace std;
int main()
{ int year,month;
  cin>>year>>month;
  switch (month)
  {case 1:cout<<31<<endl;break;
   case 3:cout<<31<<endl;break;
   case 5:cout<<31<<endl;break;
   case 7:cout<<31<<endl;break;
   case 8:cout<<31<<endl;break;
   case 10:cout<<31<<endl;break;
   case 12:cout<<31<<endl;break;
   case 4:cout<<30<<endl;break;
   case 6:cout<<30<<endl;break;
   case 9:cout<<30<<endl;break;
   case 11:cout<<30<<endl;break;
   case 2: if (year % 400==0 || year % 4==0 && year % 100!=0)
           cout<<29<<endl;
           else cout<<28<<endl;
           break;
   default:cout<<"输入错误"<<endl;
  }
  system("pause");
  return 0;
}
```

### **【练习】**

- 1、编程模拟剪刀、石头、布游戏：用 S 表示剪刀，用 R 表示石头，用 P 表示布。规则是：剪刀剪布，石头砸剪刀，布包石头。游戏者分别把自己的选择输入，计算机给出结果
- 2、输入一个字符 'A'...'Z', 'a'...'z', '0'...'9', 判断字符类型为小写字母，大写字母，数字，输入错误。如：输入 a，输出：小写字母。

---

## 七、“?”运算符

灵巧的运算符“? ”，能够用来替代某些 if-then-else 形式的语句。

这个三目的一般形式是：

```
EXP1?EXP2:EXP3;
```

现求表达式 1 的值，如果表达式 1 为真，返回表达式 2 的值；如果表达式 1 为假，返回表达式 3 的值。

典型例子 1: `max=a>b?a:b;`

```
min=a<b?a:b;
```

```
a>b?cout<<a:cout<<b;
```

```
a<b?cout<<a:cout<<b;
```

## 第四课 循环语句

### 一、For 语句

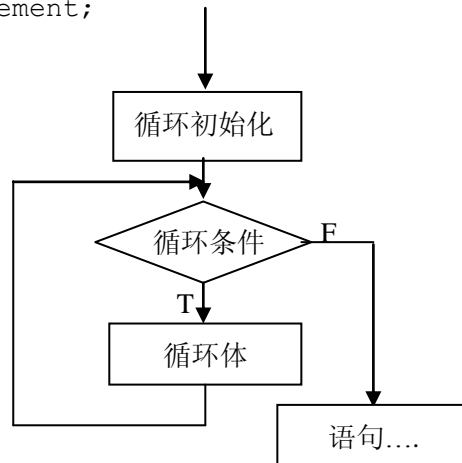
**【作用】** 所谓循环，就是不断地重复某一段程序段。

**【格式】**

```
for (init;condition;increment) statement;  
for (初始化;循环条件;增值) 循环体  
递增型 for (i=1;i<=n;i++) { 循环体 }  
递减型 for (i=n;i>=1;i--) { 循环体 }
```

**【自增、自减运算符】**

```
int i=5;  
i++; 参与运算后, i 的值自增 1  
i--; 参与运算后, i 的值自减 1  
++i; i 的值自增 1 后, 参与运算  
--i; i 的值自减 1 后, 参与运算  
i+=2; 步长为 2  
i+=5; 步长为 5
```



**【循环体验】**

**例 4.1** 请输出 5 个 '\*' 号。

```
#include <iostream>  
#define n 5  
using namespace std;  
int main()  
{ int i; //常用 I,J,K 作为循环变量的标识符  
  for (i=1;i<=n;i++) cout<<"* "; //i--控制循环次数  
  cout<<endl;  
  system("pause");  
  return 0;  
}
```

**分解循环**

```
i=1 判断<=n, 成立, 打印* 后, 执行 i++  
i=2 判断<=n, 成立, 打印* 后, 执行 i++  
i=3 判断<=n, 成立, 打印* 后, 执行 i++  
i=4 判断<=n, 成立, 打印* 后, 执行 i++  
i=5 判断<=n, 成立, 打印* 后, 执行 i++  
i=6 判断<=n, 不成立, 退出循环
```

```
#define n 10
int main()
{ int i;
  for (i=1;i<=n;i++) cout<<i<<' '; //i--用循环来控制打印的次数和内容
  cout<<endl;
  system("pause");
  return 0;
}
```

例 4.3 请输出 10-1

```
#include <iostream>
#define n 10
using namespace std;
int main()
{ int i;
  for (i=n;i>=1;i--) cout<<i<<' '; //i--用循环来控制打印的次数和内容
  cout<<endl;
  system("pause");
  return 0;
}
```

例 4.4 请输出 'A'-'Z'

```
#include <iostream>
using namespace std;
int main()
{ char ch;
  for (ch='A';ch<='Z';ch++) cout<<ch<<' '; //ch--控制循环次数和内容
  cout<<endl;
  system("pause");
  return 0;
}
```

例 4.5 请输出满足正整数，小于 n 的所有 7 的倍数 (7, 14, 21, .....)

```
#include <iostream>
#define n 5
using namespace std;
int main()
{ int i;
  for (i=0;i<=n;i+=7) cout<<i<<' ';
  cout<<endl;
  system("pause");
  return 0;
}
```

---

{循环控制变量必须是顺序类型的变量。可以是**整型**，**字符型**，可以是实型。}

请分别运行下列 2 个程序，观察结果，一样吗？

```
#include <iostream>
using namespace std;
int main()
{ float x;
  for (x=0;x!=1.0;x+=0.1) cout<<x<<" ";
  system("pause");
  return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{ float x;
  for (x=0;x<1.0;x+=0.1) cout<<x<<" ";
  system("pause");
  return 0;
}
```

## 思考原因？

例 4.5 输入  $n$ ，计算  $S=1+2+3+\dots+n$

分析：  $S=S+1$   
 $S=S+2$   
 $S=S+3$   
...  
 $S=S+n$

变量  $s$  的作用：**累加器**

以上语句可化为： $s=s+i$

在写循环时注意分析执行语句中相同的部分和不同的部分。



```
#include <iostream>
#define n 100000
using namespace std;
int main()
{ int s=0,i;
  for (i=1;i<=n;i++) s+=i;
  cout<<s<<endl;
  system("pause");
  return 0;
}
```

试一试:

- 1、将 n 的值扩大，观察结果变化，思考为什么？
- 2、将 int 改成 long long ，再扩大 n，直到 s 出现负数。

练习:

- 1、输入 n, 计算  $S=1*2*3*...*n$
- 2、输入 n, 计算  $s=2+4+6+8+...+2*n$

例 4.6 输入 n 个整数，输出其中的最大值

分析：读入一个整数：cin>>x;

读入 n 个整数：for (i=1;i<=n;i++) cin>>x; i---表示次数

```
#include <iostream>
#define n 5
using namespace std;
int main()
{ int x,max,i;
  cin>>max;
  for (i=2;i<=5;i++)
  { cin>>x;
    if (x>max) max=x;
  }
  cout<<max<<endl;
  system("pause");
  return 0;
}
```

**Max:** 哨兵的作用，用来存储当前的最大值

**例 4.7** 输出 10 个整数,统计其中正数、负数和零的个数。

分析: 用三个变量来累计个数, s1---正数的个数, s2---负数的个数, s3---零的个数

```
#include <iostream>
#define n 5
using namespace std;
int main()
{ int x,i,s1,s2,s3;
  s1=s2=s3=0;
  for (i=1;i<=n;i++)
  {
    cin>>x;
    if (x>0) s1++;
    else x==0?s2++:s3++;
  }
  cout<<s1<<' '<<s2<<' '<<s3<<endl;
  system("pause");
  return 0;
}
```

S1, S2, S3 --- 计数器  
注意: 计数器要赋初值 S1= 0

可以用 2 个?运算符解决  
x>0?s1++:x==0?s2++:s3++;

**例 4.7** 计算 fobinacci 数列{f(j), j=1, 2, 3, ...} 是这样定义的

f(0)=f(1)=1

f(n)=f(n-1)+f(n-2)

即 1, 1, 2, 3, 5, 8, 13, 21, 35, ...

输入 n, 求 fobinacci 数列的第 n 项。

## 二、For 循环的变体

### 【变体 1】

```
for (init;condition;increment) statement;
```

For 语句的三个部分的任何一个表达式都可以被省略, 它们是任选的。

例子 1: for(;;) printf("this loop will run forever.\n");

例子 2: for (x=0;x!=123;) scanf("%d",&x);

### 【变体 2】 使用逗号运算符使得两个或者两个以上的变量共同实现对循环的控制

例子 1: 对数组 s 转置到数组 t 中。

```
For (i=0,j<n;i<n;i++,j--) t[j]=s[i];
```

## 三、逗号运算符

逗号运算符用于将多个表达式串在一起, 逗号运算符的左边总是不返回的, 右边表达式的值才是整个表达式的值。

---

$x=(y=3,y+1)$ ; 相当于  $y=3,x=y+1$ ;

**练习:**

例题 4-1、(score)输入 10 个学生的成绩, 统计其中及格和不及格的人数。计算平均分。

例题 4-2、(Max1)输入 10 个整数, 输出其中的最大值

例题 4-3、(daffodil)求所有水仙花数。若三位数  $abc$ ,  $a^3+b^3+c^3=abc$ , 则称该数为水仙花数,  $153=1^3+5^3+3^3$ , 153 是水仙花数。

例题 4-4 (aabb) 输出所有形如 aabb 的四位完全平方数 (前两位相同, 后两位相同)。

例题 4-5 ( fobinacci )计算 fobinacci 数列 $\{f(j), j=1, 2, 3, \dots\}$  是这样定义的

$f(0)=f(1)=1$

$f(n)=f(n-1)+f(n-2)$

即 1, 1, 2, 3, 5, 8, 13, 21, 35, ...

输入  $n$ , 求 fobinacci 数列的第  $n$  项。

样例输入: 7 样例输出: 21

## 第五课 数组的使用

### 知识点: 声明数组:

数据类型 数组名 [第一维元素总数] [第二维元素总数]... [第一维元素总数]

例子: `int s[10];`  
`float score[3][5];`

### 一、一维数组: 数组代表的并不是一个变量, 而是一批变量。

1、声明 `int s[5];`

`float a[10];`

`char ch[20];`

`int s[5]={2,1,3,1,2};` //可以再定义变量时, 直接给初始值。

`Char ch[4]='E','W','N','S';`

`int s[5]` 等价于 `int s[0],s[1],s[2],s[3],s[4]`

说明:  $s$ :数组名, 5 为数组元素个数。

数组的每个单元:  $s[0],s[1],s[2],s[3],s[4]$

注意下标, 数组元素, 下标类型, 数组类型的区别:

数组元素:  $s[0],s[4],a[0],a[9],ch[0],ch[19],s[i],a[j],ch[i]$

对应的下标: 0 4 0 9 0 19 i j i

数组的下标是有序类型 `int`, 数据单元下标从 0 开始

数组类型是指每个单元的类型。 $s[i]$ ---`int` 类型,  $a[i]$ ---`float`,  $ch[i]$ ---字符类型

## 2、引用数据:

下标为整型: `s[3]` (使用该单元数据)    `&s[3]` (该单元地址,用于输入数据时)  
下标为变量: `s[i]` (使用该单元数据)    `&s[i]` (该单元地址,用于输入数据时)  
下标为表达式 `s[i+1],s[2*i]`            `&s[i+1],&s[2*i]`

## 3、输入数据:

通过下标访问每个单元,下标从 0 开始。

实现: `int i; //下标类型为整型`  
`For (i=0;i<5;i++) scanf("%d",&s[i])`

从键盘上输入: 5 2 1 10 14, 结果如下表:

下标	0	1	2	3	4
数组单元	<code>s[0]</code>	<code>s[1]</code>	<code>s[2]</code>	<code>s[3]</code>	<code>s[4]</code>
数据	5	2	1	10	14

## 4、输出数据

通过下标访问每个单元,下标从 0 开始。

实现: `int j; //下标类型为整型`  
`For (j=0;j<9;j++) printf("%f",&a[i]);`

## 5、常见的数组使用

```
int i,a[10];
```

1、对数组元素 `a[0]` 到 `a[9]` 的值分别为 0 到 9;

```
for (i=0;i<10;i++) s[i]=i;
```

2、数组清 0, 最常用的数据初始化的方法

```
for (i=0;i<10;i++) s[i]=0;
```

3、产生 10 个  $0 \leq s[i] < 100$  随机数,

```
srand(20101010); //找个整数做为随机种子
```

```
for (i=0;i<10;i++) s[i]=rand()%100;
```

**例 1:** 从键盘输入 `n` 个数, 将这 `n` 个数逆序输出, 并求这 `n` 个数的和, 输出这个和。

```
#include <iostream>
#define N 5
using namespace std;
int main()
{   int s,i,a[N];
```

```

for (i=0;i<N;i++) cin>>a[i];
for (i=N-1;i>=0;i--) cout<<a[i]<<' ';
cout<<endl;
s=0;
for (i=0;i<N;i++) s+=a[i];
printf("%d\n",s);
system("pause");
return 0;
}

```

**例2**、输入 n 个字母，字母为 '0'...'9'；求各字母出现的次数 (n<=255)。

分析：1、实现输入 n 个字母？

```
for (i=0;i<=n-1;i++) cin>>ch;
```

2、统计字母出现的次数？

有 10 个字母，需要 10 个计数器。 int c[10]; c[i]-统计数字 i 的个数

初始化: for (i=0;i<10;i++) c[i]=0;

算法分析: 读入字母 ch, 相应的整数为 k, c[k]++

```

#include <iostream>
#define N 5 //N-输入的字母的个数。
using namespace std;
int main()
{
    int i,c[10];
    char ch;
    for (i=0;i<10;i++) c[i]=0;//计数器清 0;
    for (i=0;i<N;i++)
    { scanf("%c",&ch); c[ch-48]++; }//每输入一个字符，相应的计数器+1;
    for (i=0;i<10;i++)
    if (c[i]!=0) cout<<i<<" "<<c[i]<<endl;//输出
    system("pause");
    return 0;
}

```

**练习**、

输入 n 个字母，字母为 'A'...'Z'；求 n 个字母中包含了几个不同的字母。

输入 n 个整数，

## 二、二维数组

二维数组元素有两个下标：第一个下标表示该元素在第几行，第二个下标表示在第几列。

1、**声明**：`int s[4][3];`

```
Int d[4][2]={{1,0},{0,1},{0,-1},{-1,0}};
```

	0 列	1 列	2 列
0 行	S[0][0]	S[0][1]	S[0][2]
1 行	S[1][0]	S[1][1]	S[1][2]
2 行	S[2][0]	S[2][1]	S[2][2]
3 行	S[3][0]	S[3][1]	S[3][2]

数据	横坐标(行)	纵坐标(列)
S[i][j]	i	j

2、**引用数据**：

下标	使用该单元数据	该单元地址
整型	s[0][0]	&s[0][0]
变量	s[i][j]	&s[i][j]

3、**输入数据**：

通过下标访问每个单元，坐标都是从 0 开始。

```
实现：int i,j; //下标类型为整型  
for (i=0;i<4;i++)  
for (j=0;i<3;j++) cin>>s[i][j];
```

4、**输出数据**

通过下标访问每个单元，下标从 0 开始。

```
实现：int i,j; //下标类型为整型  
for (i=0;i<4;i++)  
{ for (j=0;i<3;j++) cout<<s[i][j]<< " ";  
cout<<endl;  
}
```

5、**二维数组的使用**

**练习 1**、竞赛小组共有 10 位同学，每位同学共参与了三场比赛，请统计每位同学的平均分。

分析：需要一个 10 行 3 列的二维表来存放这些成绩

```
int s[10][3];s[i][j] :表示第 i 位学生，课程 j 的成绩为 s[i][j]
```

**练习 2**：输入一个 3 行 3 列整型数组，求数组中数据的最大值和最小值。

**练习 3**：输入一个 3 行 3 列的矩阵，判断该矩阵是否对称，如果没有对称就进行矩阵的转置。



**练习 4:** 输入一个 5 行 5 列的字符数组，输入#和&所在的坐标。

```

输入: #****
      *****
      *****
      *****
      *****&
    
```

输出: # (1,1)  
& (5,5)

**练习 5:** (point) 求一个行数  $n$  列数  $m$  的二维数组 ( $n*m$  个数据都不相同) 的鞍点。所谓鞍点: 数据  $s[i][j]$  , 在第  $i$  行是最大的, 在第  $j$  列是最小的, 则  $s[i,j]$  是鞍点。

输出鞍点所在的位置, 如果没有鞍点, 输出 "No!"

```

输入:
4 4
1 2 3 4
5 6 8 7
9 10 11 12
13 14 15 16
输出: 1 4
    
```

**练习 6:** (Martix) 蛇形填数

在  $n*n$  方阵中填入  $1, 2, \dots, n*n$ , 要求填成蛇形。例如  $n=4$  时方阵为:

10	11	12	1
9	16	13	2
8	15	14	3
7	6	5	4

## 第六课：条件控制循环 while、Do--while 循环

### 一、while 语句

**【作用】** 条件为 True 时，重复某一段程序段。

**【格式】** while (条件) 语句；

```
While (条件) { 语句组; }
```

当语句组的语句较少时，

```
While (条件) 语句 1, 语句 2;
```

#### 【使用体验】

**例 6.1** 请输出 5 个 '\*' 号。

```
#include <iostream>
#define n 5
using namespace std;
int main()
{ int i;
  i=1;
  while (i<=n) cout<<"* ",i++;
  cout<<endl;
  system("pause");
  return 0;
}
```

#### while 语句的使用

- 1、循环的条件 ( $i \leq n$ ),  $i$  用来控制循环的次数
- 2、循环的控制条件, 在循环外要初始化  $i=1$ ;
- 3、次数  $i$  不会自动加 1, 要有代码  $i++$

#### 分解循环

```
i=1 判断<=n , 成立, 打印* 后, 执行 i++
i=2 判断<=n , 成立, 打印* 后, 执行 i++
i=3 判断<=n , 成立, 打印* 后, 执行 i++
i=4 判断<=n , 成立, 打印* 后, 执行 i++
i=5 判断<=n , 成立, 打印* 后, 执行 i++
i=6 判断<=n , 不成立, 退出循环
```

**例 6.2** 请输出 1-10。

```
#define n 5
using namespace std;
int main()
{ int i=1;
  while (i<=n) cout<<i<<" ",i++;
  //i--用循环来控制打印的次数和内容
  cout<<endl;
  system("pause");
  return 0;
}
```



**例 6.3** 请输出 10-1

```
#define n 5
using namespace std;
int main()
{ int i=n;
  while (i>0) cout<<i--<<" ";
  //i--用循环来控制打印的次数和内容
  cout<<endl;
  system("pause");
  return 0;
}
```

**例 6.4** 请输出 'A'-'Z'

```
#include <iostream>
#define n 5
using namespace std;
int main()
{ char ch='A';
  while (ch<='Z') cout<<ch++<<" ";
  cout<<endl;
  system("pause");
  return 0;
}
```

**练习 1:** 输入一个整数，判断该数是否为素数

分析：素数的定义：只能被 1 和本身整除的数，称为素数  
x 不能被 i 整除：  $x \% I \neq 0$ ，I 的范围为 2 至  $i-1$   
利用穷举法、标识法来实现。

```
#include <iostream>
using namespace std;
int main()
{ int i,x;
  bool flag;
  cin>>x;
  flag=1; //初始化，默认 x 的素数
  for (i=2;i<=x-1;i++)
    if (x % i==0) flag=0; //只要发现被整除，x 就不是素数
  if (flag) cout<<x<<"是素数\n";
  else cout<<x<<"不是素数\n";
  system("pause");
  return 0;
}
```

优化上面的算法，检查的范围可以缩小为 2 至  $\sqrt{x}$   
只要发现 x 被某个数整除，那么就可以判断 x 不是素数。

```

#include <iostream>
#include <cmath>
using namespace std;
int main()
{ int i,x;
  cin>>x;
  i=2;
  while (i<=(int) sqrt(x) && x % i!=0) i++;
  //for (i=2; i<= (int) sqrt(x) && x %i !=0;i++);
  if (i>int(sqrt(x))) cout<<x<<"是素数\n";
  else cout<<x<<"不是素数\n";
  //i>int(sqrt(x))? cout<<x<<"是素数\n": cout<<x<<"不是素数\n";
  system("pause");
  return 0;
}

```

练习 2: 输入两个整数，求这两个数的最大公约数、最小公倍数

分析：辗转相除法。原理：最大公约数函数  $\text{gcd}(a,b)=\text{gcd}(b,a \bmod b)$

当  $a \bmod b = 0$  时， $a$  就是当前的最大公约数。最小公倍数？

```

#include <iostream>
using namespace std;
int main()
{ int a,b,t;
  cin>>a>>b;

  while (a %b!=0)  t=b,b=a % b,a=t;

  cout<<b<<endl;
  system("pause");
  return 0;
}

```

```
for (;a%b!=0;) t=b,b=a % b,a=t;
```

练习 3: 输入一个整数，素因数分解。

分析：一个正数的素因数是从 2 到 n，其中可能有多个相同的素因子，如  $8=2*2*2$ ；可能不包含某个因子，如：  $21=3*7$  ，不包含 2 和 5。从小到大，穷举因子。

```
#include <iostream>
using namespace std;
int main()
{ int x,t,i,c=0;
  int s[100];
  cin>>x; t=x;
  i=2;
  while (t!=1)
  { if (t % i==0) while (t!=1) t % i==0?s[c++]=i,t/=i:i++;
    {s[c++]=i; t/=i;}
    else i++;
  }
  cout<<x<<"="<<s[0];
  for (i=1;i<c;i++) cout<<"*"<<s[i];
  cout<<endl;
  system("pause");
  return 0;
}
```

练习 4: 输入小写字符，直到遇到字符“#”为止，统计输入的字符数和各种字符的数目。

分析：统计字符个数：如何表示？利用一个计数器来累计字符个数  
遇到# 如何表示？

```
#include <iostream>
using namespace std;
int main()
{ int i,s[26];
  char ch;
  cin>>ch;
  for (i=0;i<26;i++) s[i]=0; memset(s,0,sizeof(s));
  while (ch!='#') s[ch-97]++,cin>>ch;

  for (i=0;i<26;i++)
    if (s[i]>0) printf("%c %d\n",i+97,s[i]);

  system("pause");
  return 0;
}
```

**练习 5:** 已知一有序数列, 输入数  $x$ , 判断数  $x$  是否在数列中, 如果有, 输出其所在的位置, 否则输出“找不到”。(顺序查找、折半查找两种方法实现)。

**顺序查找:** 从数组的第一个元素往后找, 直到找到为止。找到:  $x=s[i]$

考虑特殊情况: 当指针  $i>n$  时还没找到, 就是找不到的情况。

算法有两种情况退出查找: 第一种:  $x=s[i]$ ; 第二种:  $i>n$

```
#include <iostream>
#define N 10
using namespace std;
int main()
{
    int x,i;
    int s[N]= {1,4,9,10,15,27,100,101,200,300};
    cin>>x; i=0;
    while (i<N && x!=s[i]) i++;
    if (i==N) cout<<"找不到"<<endl;
    else cout<<"第"<<i+1<<"个数"<<endl;
    system("pause");
    return 0;
}
```

请把该程序改成从后往前查找。

**折半查找:** 先确定待查的范围, 逐步缩小范围直到找到或找不到为止。

```
#include <iostream>
#define N 10
using namespace std;
int main()
{ int x,i,L,R,Mid;
  int s[N]={1,4,9,10,15,27,100,101,200,300};
  for (i=0;i<N;i++) cout<<s[i]<<" ";
  cout<<endl;
  cin>>x;
  L=0;R=N-1;Mid=(L+R)/2;
  while (L<=R && s[Mid]!=x)
  { if (s[Mid]>x) R=Mid-1; else L=Mid+1;
    Mid=(L+R)/2;
  }
  if (L>R) cout<<"找不到"<<endl;
  else cout<<"第"<<Mid+1<<"个数"<<endl;
  system("pause");
  return 0;
}
```

---

**练习 6:** 输入一个整数，计算该整数的位数及各位之和。

分析：变量：w—位数，sum—总和，

$x = x \% 10$ ，删除 x 的个位；当 x 的位数为 1 时， $x = x \% 10$  执行后  $x = 0$ ，  
所以当  $x = 0$  时，位数已经取完。

```
#include <iostream>
using namespace std;
int main()
{ int x,w,sum;
  w=0;sum=0;
  cin>>x;
  while (x>0)
  { w++; sum+=x%10; x/=10; }
  cout<<w<<" "<<sum<<endl;
  system("pause");
  return 0;
}
```

**练习：**

1、 $3n+1$  问题：证明猜想：对于任意大于 1 的自然数 n，若 n 为奇数，则将 n 变为  $3n+1$ ，否则变为 n 的一半。经过若干次这样的变换后，一定会使 n 变为 1。

例如： $3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

输入 n：输出变换的次数。  $N < 10^9$

样例输入：3

样例输出：7

2、进制转换：输入一个十进制数和整数 k，将其转化为 k 进制的数，输出该数。输入一个 k 进制的数，将其转化为十进制数。

3、(apro) 近似计算

计算  $\pi/4 = 1 - 1/3 + 1/5 - 1/7 + \dots$ ，直到最后一项小于  $10^{-6}$

---

## 第七课 多重循环—打印图形

一、打印下列图形：

```
***
***
***
```

分析：用  $i$  表示行号

用  $j$  表示列号打印一行：`for (j=1;j<=3;j++) printf("* ");`

```
#include <iostream>
using namespace std;
int main()
{ int i,j;
  for (i=0;i<3;i++)
  { for (j=0;j<3;j++) printf("* ");
    printf("\n");
  }
  system("pause");
  return 0;
}
```

增加程序的通用性：用  $n$  来存储行数，用  $m$  来存储列数。以上程序可改为：

程序：

```
#include <iostream>
#define N 3
#define M 3
using namespace std;
int main()
{ int i,j;
  for (i=0;i<N;i++)
  { for (j=0;j<M;j++) printf("* ");
    printf("\n");
  }
  system("pause");
  return 0;
}
```

这个程序就可以实现任意行和任意列的图形。

## 二、打印下列图形：

```
*  
**  
***  
****
```

分析：用  $i$  表示行号：

```
for (i=0;i<n;i++)  
{ 打印第  $i$  行;  
    Printf("\n"); {换行}  
}
```

打印第  $i$  行：

$i=1$  打印 1 个字

符

行号 $I$	0	1	2	3	4	...	$I$	...	$n$
列数 $J$	0	1	2	3	4	...	$I$	...	$n$

$i=2$  打印 2 个字符

$i=3$  打印 3 个字符

$i=4$  打印 4 个字符

**总结**：第  $I$  行——打印  $I$  个字符

程序：

```
#include <iostream>  
#define N 8  
using namespace std;  
int main()  
{ int i,j;  
  for (i=0;i<N;i++)  
  { for (j=0;j<i;j++) printf("* ");  
    printf("\n");  
  }  
  system("pause");  
  return 0;  
}
```

**关键**：在于建立外循环  $I$  与内循环  $j$  的关系

三、打印下列图形：

1	1
12	22
123	333
1234	4444

分析：用  $i$  表示行号，用  $j$  来表示列号

图形 1

列号 $j$	0	1	2	3	4	...	$I$	...	$n$
打印数字 $k$	0	1	2	3	4	...	$I$	...	$n$

图形 2

行号 $I$	0	1	2	3	4	...	$I$	...	$n$
打印数字 $k$	0	1	2	3	4	...	$I$	...	$n$

```
#include <iostream>
#define N 8
using namespace std;
int main()
{ int i,j;
  for (i=0;i<N;i++)
  { for (j=0;j<i;j++) cout<<i+1<<" ";
    printf("\n");
  }
  system("pause");
  return 0;
}
```

```
#include <iostream>
#define N 8
using namespace std;
int main()
{ int i,j;
  for (i=0;i<N;i++)
  { for (j=0;j<i;j++) cout<<j+1<<" ";
    printf("\n");
  }
  system("pause");
  return 0;
}
```



三、打印下列图形：

```

      1
     1 2
    1 2 3
   1 2 3 4
  
```

分析：可以将以上图形转换为右边的图形。通过控制每行的起始位置，就可以得到上面的图形。用  $i$  来表示行号，用  $j$  来表示列号。

```

1
1 2
1 2 3
1 2 3 4
  
```

用  $i$  表示行号，用  $j$  来表示列号

行号 $i$	0	1	2	3	4	...	$I$	...	$n-1$
列号 $j$	0	1	2	3	4	...	$I$	...	$n-1$
打印符号 $K$	1	2	3	4	5	...	$i+1$	...	$N$
行前面的空格数	40	39	38	37			$40-i$		$40-n$

总结：打印图形关键在于确定：行号与列数的关系、  
行号或列号与打印图形的关系  
行号与行前面的空格数的关系。

```

#include <iostream>
#define N 8
using namespace std;
int main()
{ int i,j;
  for (i=0;i<N;i++)
  { for (j=0;j<40-i;j++) cout<<" ";//打印前面的空格
    for (j=0;j<i;j++) cout<<j+1<<" ";//打印该行的符号
    printf("\n");//打印
  }
  system("pause");
  return 0;
}
  
```

四、完成下列练习：

<pre> * * * * * * * * * *         </pre>	<pre>       *     * * *   * * * * * * * * * * * *         </pre>	<pre>       A     B B   C C C D D D D         </pre>	<pre>       A     A B   A B C A B C D         </pre>
<pre> 1 1 2 3 1 2 3 4 5 1 2 3 4 5 6 7         </pre>	<pre>       1     123   12345 1234567         </pre>	<pre>       1     1 3 5   1 3 5 7 9         </pre>	<pre>       1     121   12321 1234321 12345654321         </pre>
<pre>       A     ABA   ABCBA ABCDCBA ABCDEDCBA         </pre>	<pre>       *     * *   * * * * * * * *   * * *     * *       *         </pre>	<pre>       *     * * *   * * * * *   * * *     *         </pre>	<pre> * * * * *   * * *     *   * * * * * * * *         </pre>
<pre> 1 0 1 0 1   1 0 1     1   1 0 1 1 0 1 0 1         </pre>	<pre> 1 0 2 0 3 0 4   1 0 2 0 3     1 0 2       1   1 0 2 1 0 2 0 3 1 0 2 0 3 0 4         </pre>	<pre> 1 0 2 0 3 0 4   1 0 2 0 3     1 0 2       1   -1 0 -2 -1 0 -2 0 -3 -1 0 -2 0 -3 0 -4         </pre>	
<pre>       6     6 6 6   5 5 5 5 5  5 5 5 5 5 5  4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3  2 2 2 2 2 2  2 2 2 2 2 2  1 1 1 1 1  1 1 1     0         </pre>	<pre>       0     1 2   3 4 5  6 7 8 9   A B C D E   F G H I J K   L M N O P Q R   S T U V W X Y Z         </pre>	<pre> 1 2 4 7 1 6 2  3 5 8 2 7 3  6 9 3 8 4  0 4 9 5  5 0 6  1 7  8         </pre>	

---

## 第八课 穷举法

**例题 1**、百鸡问题：公鸡每只 5 元钱，母鸡每只 3 元钱，小鸡 3 只 1 元钱。用 100 元钱买 100 只鸡，问公鸡、母鸡和小鸡各几只？

分析：设公鸡  $x$  只，母鸡  $y$  只，小鸡  $z$  只。

$x$  取值范围为：1-20

$y$  取值范围为：1-33

$z$  值为  $100-x-y$

穷举所有的可能性，如果当前情况下，满足  $x*5+y*3+(z/3)=100$ ，得到问题的解。

**程序：**

```
#include <iostream>
#define N 8
using namespace std;
int main()
{ int x,y,z;
  for (x=1;x<=20;x++)
    for (y=1;y<=33;y++)
      { z=100-x-y;
        if (z%3==0 && x*5+y*3+z/3==100)
          printf("公鸡=%d,母鸡=%d,小鸡=%d\n",x,y,z);
      }
  system("pause");
  return 0;
}
```

**【答案】** 三种方案：

公鸡	母鸡	小鸡
4	18	78
8	11	81
12	4	84

**例 2、** 有 4 个学生，上地理课时提出我国四大淡水湖的排序如下：

甲：洞庭湖最大，洪泽湖最小，鄱阳湖第三；

乙：洪泽湖最大，洞庭湖最小，鄱阳湖第二，太湖第三；

丙：洪泽湖最小，洞庭湖第三；

丁：鄱阳湖最大，太湖最小，洪泽湖第二，洞庭湖第三；

对于各个湖泊应处的地位，每个人只说对了-一个。

根据以上情况，编一个程序，让计算机判断各个湖泊应处在第几位。

分析：用变量 a 表示洞庭湖，b 表示洪泽湖，c 表示鄱阳湖，d 表示太湖。

a:取值范围 1-4 b:取值范围 1-4 c:取值范围 1-4 d: 10-a-b-c

甲：洞庭湖最大表示为：a=1 洪泽湖最小：b=4 鄱阳湖第三：c=3

只说对了-一个：(a==1)+(b==4)+(c==3)==1

**程序：**

```
#include <iostream>
#define N 8
using namespace std;
int main()
{ int a,b,c,d;

for (a=1;a<=4;a++)
for (b=1;b<=4;b++)
if (a!=b)
{ for (c=1;c<=4;c++)
if ((a!=c) && (b!=c))
{ d=10-a-b-c;
If( (a==1)+(b==4)+(c==3)==1 &&
(b==1)+(a==4)+(c==2)+(d==3)==1 &&
(b==4)+(a==3)==1 &&
(c==1)+(d==4)+(b==2)+(a==3)==1 )
printf("洞庭湖=%d,洪泽湖=%d,鄱阳湖=%d,太湖=%d\n",a,b,c,d);
}
}
}
system("pause");
return 0;
}
```

**练习：**

1. 编程找出六位平方数，且它的前三位和后三位都是平方数。  
[225625, 114400, 324900]、
2. 用 0-9 这 10 个数不许重复拼凑出两个自然数，让它们分别是同一个数的平方数和立方数。[69<sup>2</sup>=4761, 69<sup>3</sup>=328509]
3. 如果有一对自然数，它们的和与差是一对完全平方数，我们称这两个数为“亲和数”，例如 8 和 17 这对数：17+8=25, 17-8=9, 25 和 9 都是平方数，因此 8 和 17 是一对

---

“亲和数”，编程找出 100 以内的所有亲和数。

4. 钞票换硬币

把一元钞票换成一分、二分、五分硬币（每种至少一枚），有哪些种换法？（461 种）

5. 打印由 1, 2, 3, 4 四个数字组成的数字，并统计它的个数（注：允许四位数字中有相同的数字，如：1111, 1212, 1122 等）。

6. 阶乘之和。输入  $n$ ，计算  $s=1!+2!+3!+\dots+n!$  的末 6 位（不含前导 0）。 $n \leq 10^6$ 。

样例输入:10 样例输出: 37913

7、(hanxin) 韩信点兵

相传韩信才智过人，从不直接清点自己军队的人数，只要让士兵先后以三人一排、五人一排、七人一排地变换队形，而他每次只掠一眼队伍的排尾就知道总人数了。输入 3 个非负整数  $a, b, c$ ，表示每种队形排尾的人数（ $a < 3, b < 5, c < 7$ ），输出总人数的最小值。已知总人数不小于 10，不超过 100 人。

输入: 2 1 6 输出 41

输入: 2 1 3 输出 No Answer

8、(decimal) 分数化小数

输入一个正整数  $a, b, c$ ，输出  $a/b$  的小数形式，精确到小数点后  $c$  位。

$a, b \leq 10^6, c \leq 100$ 。

例如， $a=1, b=6, c=4$ ，应输出 0.1667

## 第九课 数据排序

### 一、冒泡排序

#### 思想:



通过一趟相临数据的比较, 可以将最大的数放在起始位置

#### 算法:

第 0 趟: 50 40 23 45 14 16    j:n-1--1    s[j]与 s[j-1]比较大小  
第 1 趟: 50 45 40 23 16 14    j:n-1--2    s[j]与 s[j-1]比较大小  
第 2 趟: 50 45 40 23 16 14    j:n-1--3    s[j]与 s[j-1]比较大小  
第 3 趟: 50 45 40 23 16 14    j:n-1--4    s[j]与 s[j-1]比较大小  
第 4 趟: 50 45 40 23 16 14    j:n-1--5    s[j]与 s[j-1]比较大小

结论: 第 i 趟, 数据 j (j 的取值范围 n-1---i+1) s[j]与 s[j-1]比较大小

#### 程序实现:

```
for (i=0;i<n;i++)  
    for (j=n-1;j>i;j--)  
        if (a[j]< a[j-1])  
            { temp=a[j];a[j]=a[j-1];a[j-1]=temp; }
```

#### 思考:

1、如果要从小到大排序

2、算法能否优化:

当第 i 趟排序过程中没有进行数据交换, 说明当前数组已经有序, 停止后面的排序工作。  
请实现算法的优化!!!

## 二、选择排序

思想：

数组中有  $N$  个数，取  $i=1, 2, 3, \dots, N-1, N$ ，每次找出  $i-n$  中最小数字的与第  $i$  个数字交换位置。

$N=7$                       80 23 50 14 16 89 20

下标	0	1	2	3	4	5	6
元素	S[0]	S[1]	S[2]	S[3]	S[4]	S[5]	S[6]
数据	80	23	50	14	16	89	20

第 0 趟：14 23 50 80 16 89 20 0-n-1 中选择最小数 14，与第 0 个数交换

第 1 趟：14 16 50 80 23 89 20 1-n-1 中选择最小数 16，与第 1 个数交换

第 2 趟：14 16 20 80 23 89 50 2-n-1 中选择最小数 16，与第 2 个数交换

第 3 趟：14 16 20 23 80 89 50 3-n-1 中选择最小数 14，与第 3 个数交换

第 4 趟：14 16 20 23 50 89 80 4-n-1 中选择最小数 16，与第 4 个数交换

第 5 趟：14 16 20 23 50 80 89 5-n-1 中选择最小数 16，与第 6 个数交换

分析：第  $i$  趟： $i--n-1$  中选择最小数，与第  $i$  个数交换

共做了几趟？

程序实现：

```
#include <iostream>
#define n 7
using namespace std;
int main()
{ int a[n],i,j,min,temp; //min--存储的是最小值的位置
  for (i=0;i<n;i++) cin>>a[i]; //输入数据
  for (i=0;i<n-1;i++)
  { min=i; //最小值的下标初始化为 i
    for (j=i+1;j<n;j++)
      if (a[j]<a[min]) min=j;
    temp=a[min];a[min]=a[i];a[i]=temp; //交换
  }
  for (i=0;i<n;i++) cout<<a[i]<<" ";
  cout<<endl;
  system("pause");
  return 0;
}
```





---

<http://218.4.165.132:8080/oj/Problems>