18 函数

主教练：党东

■ 函数的定义

■ C++是由函数构成的，**函数是C++的基本模块**。有的函数完成某一操作；有的函数计算出一个值。通常，一个函数即能完成某一特定操作，又能计算数值。

■ 为什么要使用函数?

■ 1、避免重复的编程。

■ 2、使程序更加模块化，便于阅读、修改。

■ 说明：

■ 1、一个源程序文件由一个或多个函数组成。

■ 2、一个程序必须有且只有一个main( )函数，C++从main( )函数开始执行。

■3、从使用角度来说，分标准函数和用户自定义函数；

从形式来说，分无参函数和有参函数。

函数（函数定义的一般形式）

■ 无参函数

- 主调函数并不将数据传给被调函数。
- 无参函数主要用于完成某一操作。


```
类型说明    函数名（）
{   函数体   }
```

不传递参数

函数（函数定义的一般形式）

```
int main( )
{   printstar ();
    print_message ();
    printstar();
}

void  printstar ()
{    cout<< "* * * * * * * * * * *\n" ;}
void  print_message ()
{   cout<< " How do you do! \n" ;}
```

调用函数

调用函数

调用函数

函数类型

函数名

函数体

两个被调函数主要用于完成打印操作。

函数（函数定义的一般形式）

```
#include <iostream>
using namespace std;
void print_message ()
{    cout<<" How do you do! \n";}
void printstar ()
{    cout<<"* * * * * * * * * * *\n";}
int main( )
{   printstar ();
     print_message ();
     printstar();
     return 0;
}
```

```
#include <iostream>
using namespace std;
int main( )
{   cout<<"* * * * * * * * * * *\n";
     cout<<" How do you do! \n";
     cout<<"* * * * * * * * * * *\n";
      return 0;
}
```

输出：* * * * * * * * * * * *

How do you do!

* * * * * * * * * * * *

函数（函数定义的一般形式）

■ 有参函数

■ 主调函数和被调函数之间有数据传递。

■ 主调函数可以将参数传递给被调函数，被调函数中的结果也可以带回主调函数。

```
类型说明  函数名（形式参数列表说明）
{  函数体  }
```

函数（函数定义的一般形式）

```
int  max (int x,int y)
{   int  z;
    if(x>y)z=x;else z=y ;
    return  z;
}
```

函数类型

形参列表说明

函数名

函数体

函数值

```
int main ()
{   int   a,b,c;
    cin>>a>>b;
    c=max (a , b) ;
    cout<< “The max is” << c<<endl;
}
```

主函数

调用函数

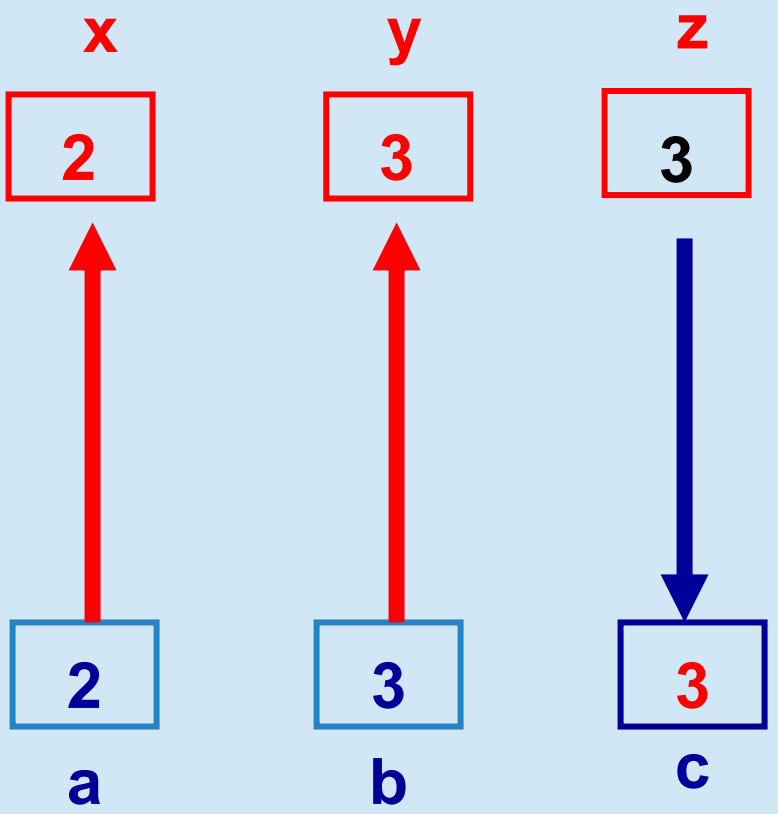
实际参数

将实际值a,b传给被调函数的参数x,y，计算后得到函数值z返回

函数（函数定义的一般形式）

```
int  max (int x,int y)
{   int  z;
    if(x>y)z=x;else z=y;
    return  z;
}

int main ( )
{   int   a,b,c;
    cin>>a>>b;
    c=max (a , b) ;
    cout<< “The max is” << c<<endl;
}
```



■ 函数参数和函数的值

- 形参是被调函数中的变量；实参是主调函数赋给被调函数的特定值。实参可以是常量、变量或复杂的表达式，不管是哪种情况，在调用时实参必须是一个确定的值。
- 形参与实参类型相同，一一对应。
- 形参必须要定义类型。

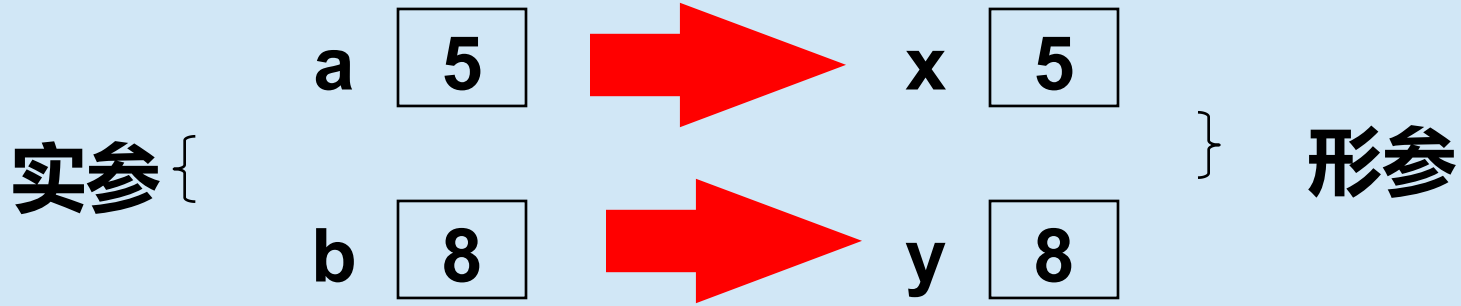
```
int  max (int x,int y)
{   int  z;
    if(x>y)z=x;else z=y;
    return  z;
}
int main ( )
{    int   a,b,c;
     cin>>a>>b;
     c=max (a+b , a*b) ;
     cout<< “The max is” <<c<<endl;
}
```

若a为3，b为5，则实参为8, 15，分别送给形参x，y。

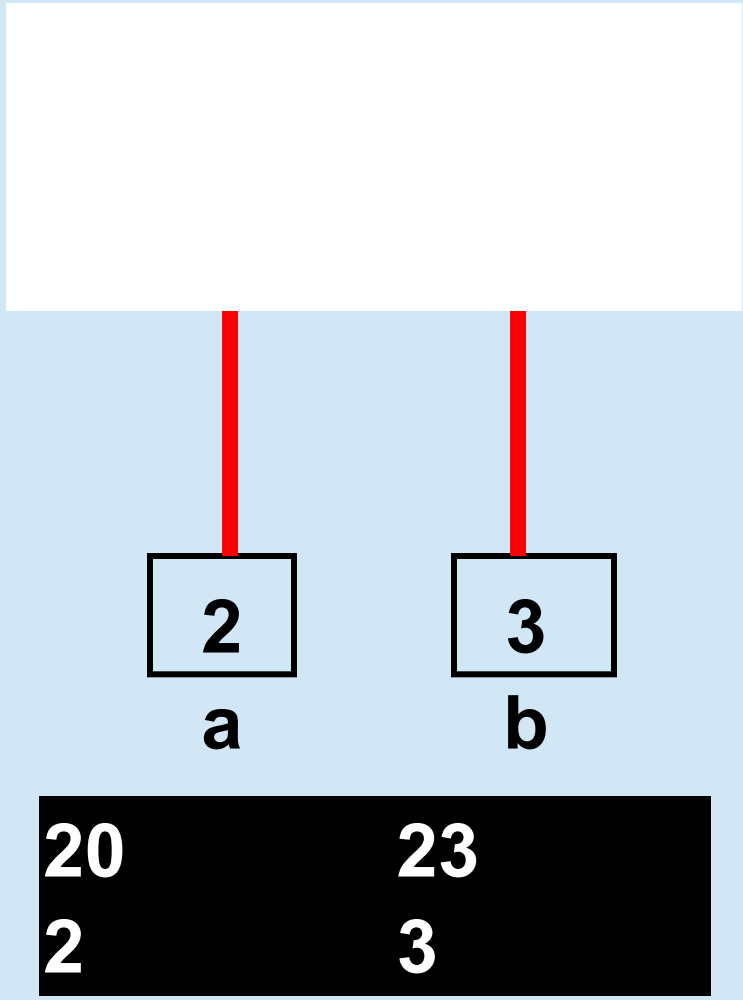
先计算，后赋值

■ 说明：

- 1、在未出现函数调用时，形参并不占内存的存储单元，只有在函数开始调用时，形参才被分配内存单元。调用结束后，形参所占用的内存单元被释放。
- 2、实参对形参变量的传递是“值传递”，即单向传递。在内存中实参、形参分别占不同的单元。
- 3、形参只作用于被调函数，可以在别的函数中使用相同的变量名。



```
void fun(int a, int b)
{    a=a*10;
     b=b+a;
     cout<<a<<  '\t'  <<b<<endl;
}
int main( )
{   int a=2, b=3;
    fun(a,b);
    cout<<a<<  '\t'  <<b<<endl;
}
```



■ 函数的返回值

- 函数的返回值通过return语句获得。函数只能有唯一的返回值。
- 函数返回值的类型就是函数的类型。
- return语句可以是一个表达式，函数先计算表达式后再返回值。
- return语句还可以终止函数，并将控制返回到主调函数。
- 一个函数中可以有一个以上的return语句，执行到哪一个return语句，哪一个语句起作用。

■ 函数的返回值

```
int add ( int  a, int b)
{
    return (a+b);
}
```

先计算，后返回

若函数体内没有return语句，就一直执行到函数体的末尾，然后返回到主调函数的调用处。

```
int max ( int a, int b)
{
    if (x>y) return   x ;
    else  return  y;
}
```

可以有多个return语句

■ 函数的返回值

■ 既然函数有返回值，这个值当然应属于某一个确定的类型，应当在定义函数时指定函数值的类型。

```
int   max (double a, double  b) // 函数值为整型
```

函数返回值的类型，也是函数的类型

■ 不带返回值的函数可说明为void型。

■ 函数的类型与函数参数的类型没有关系。

```
double   blink ( int a,  int b)
```

■ 如果函数的类型和return表达式中的类型不一致，则以函数的类型为准。函数的类型决定返回值的类型。对数值型数据，可以自动进行类型转换。

■ 函数的返回值

```
int max ( int a, int b)
{   int z;
    if(x>y)z=x;else z=y;
    return  z;
}
```

如果有函数返回值，函数的类型就是返回值的类型

如果有函数返回值,返回值就是函数值,必须惟一。

函数体的类型、形式参数的类型必须在函数的定义中体现出来。

参数（多个） → 函数体 → 函数值(唯一)

■ 数组名作函数参数

- 数组名可以作函数的实参和形参，传递的是**数组的地址**。这样，**实参、形参共同指向同一段内存单元**，内存单元中的数据发生变化，这种变化会反应到主调函数内。

- 在函数调用时，**形参数组并没有另外开辟新的存储单元**，而是以实参数组的首地址作为形参数组的首地址。**这样形参数组的元素值发生了变化也就使实参数组的元素值发生了变化**。

■ 形参实参都用数组名

```
int main( )
{  int  array[10];
   ......
   f(array, 5);
   .....
}
```

实参数组

形参数组,必须进行类型说明

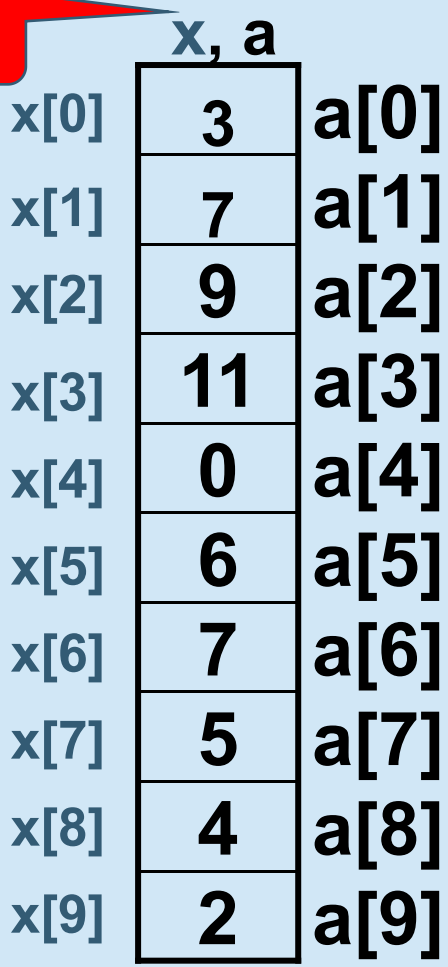
```
Void f(int  arr[ ], int n)
 {
        ......
}
```

用数组名作形参，因为接收的是地址，所以可以不指定具体的元素个数。

■ 将数组中的N个数按从小到大输出。

```
void  sort(int x[ ])
{   int  t, i, j;
    for (i=0;i<=8; i++)
        for(j=i+1;j<=9;j++)
            if(x[i]>x[j]){t=x[i];x[i]=x[j];x[j]=t;}
}
int main( )
{    int  i, a[10]={0};
     for(i=0;i<10;i++)cin>>a[i];
     sort(a);
     for (i=0;i<10; i++)cout<<a[i]<< “  ” ;
}
```

x与a数组指向同一段内存



■ 函数的调用

■ 函数调用的一般形式

函数名（实参列表）；

■ 形参与实参类型相同，一一对应。

■ 函数调用的方式

1.作为语句　　printstar( );

2.作为表达式　　c=max (a,b);

3.作为另一个函数的参数　　cout<<max (a,b);

注： 函数调用遵循先定义、后调用的原则，即被调函数应出现在主调函数之前。

■ 函数的调用

```
double max(double x, double y)
{   double z;
    if(x>y)z=x;z=y;
    return  z;
}
```

形参必须说明参数类型

被调函数先定义

```
int main ()
{  double a,b, c;
   cin>>a>>b;
   c=max (a+b , a*b) ;
   cout<< “The max is” <<c<<endl;
}
```

定义之后再调用

实参传递的是一个具体的值，不必说明参数类型

【例1】求最大值 --1098

Description

已知m=max(a,b,c)/(max(a+b,b,c)*max(a,b,b+c))，其中max(a,b,c)是求三个数a,b,c的最大值；

Input

输入三个整数a,b,c, 中间用空格分开

Output

求m的值(保留三位小数)

【样例输入】

1 1 1

【样例输出】

0.250

【例2】验证哥德巴赫猜想 --1099

Description

哥德巴赫是一个古老而著名的数学难题。就是任一充分大的偶数，可以用两个素数之和表示。例如

4=2+2 6=3+3

8=3+5 98=19+79

它的理论证明很麻烦，现在我们对有限范围内的数，用计算机加以验证。

试编写程序，以实验数据写出实验结果。

Input

输入一个数n（n<=10000）

Output

输出仅一对满足条件的表达式，并且小数在前，大数在后面的形式

Sample Input

98

Sample Output

19+79

【练习1】回文数 --1100

Description

编程求出n以内的，二进制和十进制正读和反读都一样的整数。

Input

输入n(n<=30000)

Output

输出满足条件的个数。

Sample Input

3

Sample Output

2

【练习2】回文数弱版 --1101

Description

任给一个三位数abc（10进制），算出abc与cba之和。若该数不是回文数（即从左到右读与从右到左读是同一个数如121），再按照上述方法求和，以此类推，直到得到回文形式的和数或者和数位数超过了15位时中止计算。

Input

输入一个三位数

Output

输出如能够满足要求的回文数，输出加的次数，否则输出“Fail!”

Sample Input

123

Sample Output

1