实验六、函数

班级：机设（3）班 学号：115040100304 姓名：章文瑾

1. 目的要求
2. 熟练掌握函数的定义和调用方法。
3. 熟练掌握函数实参与形参的对应关系，以及“值传递”的方式。
4. 熟练掌握函数的嵌套调用和递归函数编写的规律。
5. 学会全局变量和局部变量、动态变量和静态变量的概念和使用方法。
6. 实验步骤与内容
7. 函数的定义、声明、调用。
8. 形参和实参的概念和传递。
9. 函数的类型和返回值的类型。

4、变量的存储形式和作用范围。

1. 实验小结
2. 在使用自定义函数时，有三个步骤：函数定义、函数原型声明、函数调用。其中，函数定义就是把一个子任务程序写成一个函数，一般形式如下：

类型 函数名（形参）

{

局部变量

语句

}

注意，这里的类型可以是int、double等、也可以是空类型，即void。形参也可以不写，即无参函数。这种函数常被用来执行一组指定的操作。无参函数多是void型，但也有例外，但空类型居多。例如：

void print()

{

cout<<"C"<<endl;

}

这就是一个空类型且无参的函数，调用它只是用来输出一个“C”，在主调和被调之间没有数据的传递，也不返回任何数值。

1. 大部分的函数都是有参数且有返回值的。定义函数时的参数是形参，不调用它时，是不被赋予储存空间的。只有在调用该函数时，才被临时分配一个存储空间，但在调用结束后，这个存储空间又会被收回。将实参赋予到该函数中时，只是让形参有了一个相等的值并参与计算，所以实参实际上只是把值给了形参，让它去计算，而实参本身并不参与。所以无论在函数经历了怎样的计算，实参的值是不改变的。只有在后面的指针的使用中，才会通过形参也间接改变了实参的值，但在这里不作过多讨论。
2. 在“{}”之中的部分是函数体，它包括变量的定义与执行语句的序列。若函数有返回值，则需要返回语句return，return语句也不是必须的，一般不需要返回值的函数都是空类型。如上机题3中：

void fun(int,int,int)

{

......

cout<<"t="<<t<<endl;

}

这里调用该函数后，想直接执行输出一个t的语句。所以并不需要返回值。故这里成空类型。

1. 在上机题4中，定义了一个函数power，因为这里有返回值，所以函数不再被定义成void，而是float。这里定义的函数类型，其实就是定义返回值的类型。如果在函数体中定义的返回值类型与函数类型不一致时，则会发生冲突。若是数值型数据，则会强制转换成函数类型。若一个是字符，一个是数值型，那么程序就会扳错了。
2. 在函数使用之中，一般都用到一些变量，这就涉及到变量的作用域以及生存期等问题。变量的作用域即是变量的有效使用范围，在C语言中，变量按使用的范围可分成局部和全局两种。局部变量是在语句块内定义的变量。它的作用范围仅限于语句块内。例如在上机题5中，在主函数main与自定义函数change中，都定义了整型变量a、b、c，但是由于这两个函数中的a、b、c的作用范围都只限于各自的语句体中，所以可以有值的传递，但是，不会有超出语句体之外的影响。
3. 有时候又希望变量的作用范围不仅限于小的语句体之中，那么就要用到全局变量。该种类型的变量是在函数体语句之外定义。后面所有的函数都可以直接运用，无需再进行局部变量的定义了。
4. 变量的分类形式，除了全局与局部之外，还有自动变量、静态变量之分。未加特殊说明的变量即是自动变量。自动变量属于动态存储。即生存的周期只限于其所属的语句体内。而静态变量的生存周期则扩展到了整个程序，但在这里需要注意的是它的作用域仍与自动变量相同。例如以下函数：

void f()

{

static int j=0

++j;

cout<<j<<endl;

}

若运行这个函数三次，则输出结果依次为“1、2、3”、因为静态局部变量保存了前次被调用后留下的值。因此，j的值是依次增加的。

1. 宏定义是一种使用起来很像函数的操作，它通常在程序的最开头定义。通常用在定义某个数值时使用，例如：

#define p 3.1415926

它并不是普通的语句，所以不以分号结束。作了上面的定义之后，在之后程序的运行过程中，若想使用3.1415926这个数值，就可以直接使用p来代替，免去了多次输入的麻烦。另外宏还有一种带参数的形式，例如：

#define a(r) (3+r)

那么后面若写“a(2)”那么程序就会计算“3+2”的值。虽然这与函数的使用很像，但宏与函数在运行机制与原理上是不同的，要注意区分。