

第5章

函数

5.1 实验目的

- (1) 掌握定义函数的方法。
- (2) 掌握函数实参及形参的对应关系以及“值传递”方式。
- (3) 掌握函数的嵌套调用和递归调用的方法。
- (4) 掌握全局变量和局部变量,动态变量、静态变量的概念和使用方法。
- (5) 学会对多文件程序的编译和运行。
- (6) 掌握指针与函数的关系。
- (7) 掌握数组作为函数参数的作用。
- (8) 了解指向函数的指针。

5.2 相关知识

函数是 C 源程序的基本模块,C 语言中的函数相当于其他高级语言的子程序。C 语言不仅提供了极为丰富的库函数(如 Turbo C,MS C 都提供了三百多个库函数),还允许用户建立自己定义的函数。用户可把自己的算法编成一个个相对独立的函数模块,然后用调用的方法来使用函数。可以说 C 程序的全部工作都是由各式各样的函数完成的,所以也把 C 语言称为函数式语言。

5.2.1 函数定义的一般形式

1. 无参函数的定义形式

```
类型标识符 函数名()  
{ 声明部分  
  语句  
}
```

类型标识符指明了本函数的类型,函数的类型实际上是函数返回值的类型。该类型标识符与前面介绍的各种说明符相同。函数名是由用户定义的标识符,函数名后有一个空括号,其中无参数,但括号不可少。

{ } 中的内容称为函数体。在函数体中声明部分,是对函数体内部所用到的变量的类型说明。

在很多情况下都不要要求无参函数有返回值,此时函数类型符可以写为 void。

2. 有参函数定义的一般形式

```
类型标识符  函数名(形式参数表列)
{声明部分
  语句
}
```

有参函数比无参函数多了一个形式参数表列。在形参表中给出的参数称为形式参数,它们可以是各种类型的变量,各参数之间用逗号间隔。在进行函数调用时,主调函数将赋予这些形式参数实际的值。形参既然是变量,必须在形参表中给出形参的类型说明。

5.2.2 函数调用的方法

主调函数使用被调函数的功能,称为对被调函数的调用。函数调用的形式是通过函数名和函数的参数。

按照函数在主调函数中的作用,函数的调用方式可以有以下三种形式。

1. 函数语句

被调函数在主调函数中,以语句的方式出现,称为函数调用语句或函数语句。通常只完成一种操作,不带回返回值。

2. 函数表达式

将函数的调用结果作运算符的运算分量,这种函数是有返回值的。

3. 函数参数

函数的调用结果进一步作其他函数的实参,这种函数也是有返回值的。

5.2.3 数据在函数之间的传递

在程序中,如果仅用函数代替一个语句序列,那么函数的作用就不大了。一般情况下,常常要求同一个函数可以根据不同的数据,进行相同的处理之后得到不同的结果,这样,函数与函数之间通常要传递数据和计算结果。C语言中采用参数、返回值和全局变量三种方式进行数据传递。主调函数与被调函数之间是双向传递数据,当调用函数时,通过

函数的参数,主调函数为形参提供数据,调用结束时,被调函数通过返回语句将函数的运行结果(称为返回值)带回主调函数中,函数之间还可以通过使用全局变量,在一个函数内使用其他函数中的某些变量的结果。

1. 形参与实参

形参是函数定义时由用户定义的形式上的变量,实参是函数调用时,主调函数为被调函数提供的原始数据。

C语言函数参数采用“值传递”的方法,其含义是:在调用函数时,将实参变量的值取出来,赋值给形参变量,使形参变量在数值上与实参变量相等。在函数内部使用从实参中传递来的值进行处理。C语言中的实参可以是一个表达式,调用时先计算表达式的值,再将结果(值)赋值给形参对应的存储单元中,一旦函数执行完毕,系统将释放这些与形参对应的存储单元,这些存储单元所保存的值也不再保留。形式参数是函数的局部变量,仅在函数内部才有意义,但也可能用它来返回函数操作的结果。

值传递的优点在于被调用的函数不可能改变主调函数中变量的值,而只能改变它的局部的临时副本。这样就可以避免被调用函数的操作对调用函数中的变量可能产生的副作用。

C语言中,在“值传递”方式下,既可以在函数之间传递“变量的值”,也可以在函数之间传递“变量的地址值”。

2. 函数的返回值

函数调用之后的结果称为函数的返回值,通过返回语句带回主调函数。

1) 函数的返回语句

格式: `return 表达式;`

`return(表达式);`

功能: 将表达式的值带回主调函数。

2) 关于返回语句的说明

(1) 函数的返回值只能有一个。

(2) 当函数中不需要指明返回值时,可以写成“`return;`”,也可以不写。函数运行到右花括号自然结束。

(3) 一个函数体内可以有多个返回语句,不论执行到哪一个,函数都结束,回到主调函数。但在结构化程序设计中不提倡在一个函数中使用多个 `return` 语句。

(4) 当函数没有指明返回值,即“`return;`”或没有返回语句时,函数执行后实际上不是没有返回值,而是返回一个不确定的值,有可能给程序带来某种意外的影响。因此,为了保证函数不返回任何值,C语言规定,可以定义无类型函数,其形式为:

`void 函数名(形参表)`

`{ ... }`

`void` 类型又称为无值类型(或空类型)。首先,在概念上必须明确: `void` 类型的函数

不是调用函数之后不再返回,而是调用函数在返回时没有返回值。void 类型在 C 语言中有两个用途:一是表示一个函数没有返回值,二是用来指明一个通用型的指针。第二种用途暂不讨论。

void 类型的函数与有返回值类型的函数在定义过程中没有区别,只是在调用时不同,有返回值的函数可以将函数调用放在表达式的中间,将返回值用于计算,而 void 类型的函数不能将函数调用放在表达式之中,只能在语句中单独调用。

void 类型的函数一般用于完成一些规定的操作,而调用函数本身不再对被调用函数的执行结果进行干预(运算)。

3. 关于函数返回值的类型

函数定义时的类型就是函数返回值的类型。从理论上,C 语言要求函数定义的类型应当与返回语句中表达式的类型保持一致。当两者不一致时,系统自动进行转换,将函数返回语句中表达式的类型转换为函数定义时的类型。

5.2.4 对被调函数的声明和函数原型

由于 C 语言可以由若干个文件组成,每一个文件可以单独编译,因此在编译程序中的函数调用时,如果不知道该函数参数的个数和类型,编译系统就无法检查形参和实参是否匹配,为了保证函数调用时,编译程序能检查出形参和实参是否满足类型相同、个数相等,并由此决定是否进行类型转换,必须为编译程序提供所用函数的返回值类型和参数的类型、个数,以保证函数调用成功。这里提出函数声明的概念。

1. 函数的声明

主调函数调用被调函数之前,必须对被调函数作声明,其形式是:

函数类型 函数名(形参类型 1 形参名 1,形参类型 2 形参名 2 ...);

目的是告诉编译系统,函数返回值是什么类型,有多少个参数,每一个参数是什么类型的,为编译系统进行类型检查提供依据。

这里应当提醒的是,函数的声明和函数的定义形式上类似,但两者上有本质上的不同。

(1) 函数的定义是编写一段程序,除上面的内容之外,应有函数具体的功能语句,即函数体;而函数的声明仅是对编译系统的一个说明,不含具体的执行动作。

(2) 在程序中,函数的定义只能有一次,而函数的声明可以有几次,调用几次该函数,就应在各个主调函数中各自声明。

2. 函数原型的概念

在对被调用的函数进行声明时,编译系统需知道被调函数有几个参数,各自是什么类型,而参数的名字是无关紧要的,因此,对被调函数的声明可以简化为:

函数类型标识符 函数名(形参类型 1,形参类型 2, ...);

上述方式称为函数的原型。通常将一个文件中需调用的所有函数原型写在文件的开始。

5.3 实验内容

5.3.1 验证性实验

实例 1 函数的简单应用。

```
#include <stdio.h>
#include <math.h>
double func(double x,double y) /* 用户定义函数 */
{
    return (pow(x,y));        /* pow()为库函数 */
}
void main( )
{
    double a,b,c,d;
    scanf("%lf,%lf",&a,&b);
    c = func(a,b);           /* 第一次调用用户函数 func() */
    d = func(c,b);           /* 第二次调用用户函数 func() */
    printf("%lf,%lf\n", c, d);
}
```

程序释疑：

本例中共包含两个函数,主函数 main() 和用户定义函数 func(),主函数可以定义在程序的任意位置。在主函数中,两次调用用户函数,分别以不同的数据求出不同的结果。在用户函数 func() 中使用了系统库函数 pow()。

实例 2 写一个判断整数是否为素数的函数,并使用该函数求 1000 以内的素数平均值。

```
#include <stdio.h>
#include <math.h>
main( )
{ int a = 0, k;           /* a 保存素数之和 */
  float av;              /* av 保存 1000 以内素数的平均值 */
  for(k = 2; k <= 1000; k++)
  if ( fun(k) )          /* 判断 k 是否为素数 */
    a += k;
  av = a/1000;
```

```
printf("av = %f\n", av);
}
fun(int n) /* 判断输入的整数是否为素数 */
{ int i, y = 0;
  for(i = 2; i < n; i++)
    if(n % i == 0) y = 1;
    else y = 0;
  return y;
}
```

程序释疑：本题调试的重点是如何判断一个数是否为素数。

实例 3 定义一个函数 s ，该函数的功能是求 $\sum_{i=1}^n i$ 的值。

```
#include <stdio.h>
#include <math.h>
main()
{
    int n;
    printf("input number\n");
    scanf("%d", &n);
    s(n);
    printf("n = %d\n", n);
}
int s(int n)
{
    int i;
    for(i = n - 1; i >= 1; i--)
        n = n + i;
    printf("n = %d\n", n);
    return n;
}
```

程序释疑：

本程序定义了一个函数 s ，其功能是求 $\sum_{i=1}^n i$ 的值。在主函数中输入 n 值，并作为实参，在调用时传送给 s 函数的形参 n （注意，本例的形参变量和实参变量的标识符都为 n ，但这是两个不同的量，各自的作用域不同）。举例说，假若输入 n 值为 100，即实参 n 的值为 100，把此值传给函数 s 时，形参 n 的初值也为 100，在执行函数过程中，形参 n 的值变为 5050。返回主函数之后，输出实参 n 的值仍为 100。可见实参的值不随形参的变化而变化。

实例 4 复合语句中使用局部变量。

```
main()
{
```

```

int i = 2, j = 3, k;
k = i + j;
{
    int k = 8;
    printf(" % d\n", k);
}
printf(" % d\n", k);
}

```

程序释疑:

本程序在 main 中定义了 i, j, k 三个变量, 其中 k 未赋初值。而在复合语句内又定义了一个变量 k, 并赋初值为 8。应该注意这两个 k 不是同一个变量。在复合语句外由 main 定义的 k 起作用, 而在复合语句内则由在复合语句内定义的 k 起作用。因此程序第 4 行的 k 为 main 所定义, 其值应为 5。第 7 行输出 k 值, 该行在复合语句内, 由复合语句内定义的 k 起作用, 其初值为 8, 故输出值为 8, 第 9 行输出 i, k 值。i 是在整个程序中有效的, 第 7 行对 i 赋值为 3, 故输出也为 3。而第 9 行已在复合语句之外, 输出的 k 应为 main 所定义的 k, 此 k 值由第 4 行已获得为 5, 故输出也为 5。

实例 5 外部变量与局部变量同名的例子。

```

int a = 3, b = 5;           /* a, b 为外部变量 */
max(int a, int b)         /* 形参 a, b 为局部变量 */
{ int c;
  c = a > b ? a : b;
  return(c);
}
main()
{ int a = 8;
  printf(" % d\n", max(a, b));
}

```

程序释疑:

如果同一个源文件中, 外部变量与局部变量同名, 则在局部变量的作用范围内, 外部变量被“屏蔽”, 即它不起作用。

5.3.2 基础练习实验

(1) 分析并写出下列程序的运行结果, 然后运行此程序, 思考为什么会得到这样的结果。

① 下面程序的运行结果是()。

```

#include <stdio.h>
int a = 5, b = 7;
int plus(int, int);

```

```
void main()
{   int a = 4, b = 5, c;
    c = plus(a, b);
    printf("a + b = %d\n", c);
}
int plus(int x, int y)
{   int z;
    z = x + y;
    return(z);
}
```

② 下面程序的运行结果是()。

```
#include <stdio.h>
int s = 10;
func(int a)
{ int c;
  c = a + s;
  return c;
}
void main()
{ int x = 8, y = 9, t;
  t = func((x + y, x--, y++));
  printf("%d\n", t);
}
```

③ 下面程序的运行结果是()。

```
#include <stdio.h>
void t(int x, int y, int cp, int dp)
{ cp = x * x + y * y;
  dp = x * x - y * y;
}
void main()
{ int a = 4, b = 3, c = 5, d = 6;
  t(a, b, c, d);
  printf("%d %d\n", c, d);
}
```

④ 下面程序的运行结果是()。

```
#include <stdio.h>
void fun()
{ static int a = 0;
  a += 2;
  printf("%d", a);
}
```

```
void main()
{ int cc;
  for(cc = 1;cc < 4;cc++) fun();
  printf("\n");
}
```

⑤ 下面程序的运行结果是()。

```
# include <stdio.h>
int k = 1;
void main()
{
    void fun(int m);
    int i = 4;
    fun(i);
    printf ("\n % d, % d", i, k);
}
void fun(int m)
{
    m += k; k += m;
    {
        char k = 'B';
        printf ("\n % d", k - 'A');
    }
    printf ("\n % d, % d", m, k);
}
```

⑥ 下面程序的运行结果是()。

```
# include <stdio.h>
void func(int i, int j)
{
    int a = 3;
    static int b = 3;
    a++;
    b++;
    printf (" % d --- a = % d\n", i, a);
    printf (" % d --- b = % d\n", j, b);
}
void main()
{
    int i;
    for(i = 0; i < 3; i++)
        func(i, i++);
}
```

⑦ 下面程序的运行结果是()。

```
#include <stdio.h>
void main()
{
    int a=2,i,f(int a);
    for(i=0;i<3;i++) printf("%4d\n",f(a));
}
int f(int a)
{ int b=0;static int c=3;
  b++;c++;
  return(a+b+c);
}
```

⑧ 下面程序的运行结果是()。

```
#include <stdio.h>
#define MUL(x,y) (x)*y
void main()
{
    int a=3,b=4,c;
    c=MUL(a++,b++);
    printf("%d\n",c);
}
```

(2) 程序填空题。请按功能完善下列程序并上机调试运行。

① 下面程序的功能是：求 $S=1!+2!+3!+4!+5!$ 的值。

```
#include <stdio.h>
long fun(int i)
{
    long p=1;
    int k;
    for (k=1; _____;k++)
        p=p*k;
    return p;
}
void main()
{
    int i;
    long sum=0;
    for(i=1;i<=5;i++)
        sum=sum+ _____;
    printf("sum=%ld\n",sum);
}
```

② 下面程序的功能是：计算下面函数的值。

$$f(x, y, z) = \frac{\sin(x)}{\sin(x-y) \times \sin(x-z)} + \frac{\sin(y)}{\sin(y-z) \times \sin(y-x)} + \frac{\sin(z)}{\sin(z-x) \times \sin(z-y)}$$

```
#include <math.h>
#include <stdio.h>
void main()
{
    double f(float, float, float), sum;
    float x, y, z;
    scanf("%f %f %f", &x, &y, &z);
    sum = _____
    printf("sum = %lf\n", sum);
}
double f(float a, float b, float c)
{ double u;
  u = sin(a)/(sin(b) * sin(c));
  return(u);
}
```

③ 下面程序的功能是：从键盘输入两个正整数 a 和 b, 用辗转相除法求它们的最大公约数。

```
#include <stdio.h>
int hcf(int m, int n)
{
    int r;
    if(m < n)
    { r = m; _____; n = r; }
    r = m % n;
    while(_____)
    {
        m = n;
        n = r;
        r = m % n;
    }
    _____;
}
void main()
{
    int a, b, c;
    scanf("%d %d", &a, &b);
    c = hcf(_____);
}
```

```
    printf(" %d\n",c);  
}
```

④ 下面程序的功能是：求 101~200 之间有多少个素数，并输出所有素数（每行 5 个数）。

```
#include <stdio.h>  
#include <math.h>  
int isprime(int n)  
{    int i;  
    for(i = 2; i <= sqrt(n); i++)  
        if(n % i == 0)  
            _____;  
    return 1;  
}  
void main()  
{    int i,k = 0,num = 0;  
    for(i = 101; _____; i++)  
        if(isprime(i) == 1)  
            {    num++;  
                printf(" %4d", i);  
                if(num % 5 == 0)printf("\n");  
            }  
    printf("\nThe total is %d\n", num);  
}
```

(3) 程序改错。请按照下列各程序的功能修改程序并上机调试运行。

① 下面程序的功能是：用函数实现计算 $n!$ 。

```
#include <stdio.h>  
void main()  
{  
    double f(int);  
    int n;  
    scanf(" %d", &n);  
    printf("n = %d, f = %f\n", n, f(n));  
}  
double f(int n)  
{ double u = 1.0;  
    while(n > 1 && n < 170) u * = (double)n-- ;  
    return(u);  
}
```

② 下面程序的功能是：利用公式 $e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{n!}$ 求 e 的近似值。

```
#include <stdio.h>
```

```
long fact(int n)
{
    long i = 1;
    i = i * n;
    return i;
}
void main()
{
    long e, p ;
    int i = 0;
    e = 0;
    p = 1;
    while(p >= 0.00001)
    {
        i = i + 1;
        p = 1/fact(i);
        e = p;
    }
    printf("e = %7.3f\n", e);
}
```

③ 下面程序的功能是：计算学生的年龄。已知第一位最小的学生年龄为 10 岁，其余学生的年龄一个比一个大两岁，求第 5 个学生的年龄。

```
# include <stdio.h>
int age( int n )
{
    int c;
    if( n == 1)
        c = 10;
    else
        c = 12;
    return(c);
}
int main()
{
    int n = 5;
    printf("age: %d\n", age(5) );
}
```

5.3.3 设计性实验

(1) 有变量定义和函数调用语句：`int a=1,b=-5,c;c=fun(a,b);`，`fun` 函数的作用是计算两个数之差的绝对值，并将差值返回调用函数。请编写该函数。

```
int fun(int x, int y)
{      }
```

(2) 写一个函数,计算 x 的 n 次方。

(3) 有变量定义和函数调用语句: `int x=57;,prime(x);`,`prime` 函数用来判断一个整数 a 是否为素数,若是素数则函数返回 1,否则函数返回 0。请编写该函数。

```
int prime(int a)
{      }
```

(4) 以下程序的功能是应用下面的近似公式计算 e 的值。函数 `f1` 用来计算每项分子的值,函数 `f2` 用来计算每项分母的值。请编写 `f1` 和 `f2` 函数。

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots (\text{前 20 项的和})$$

```
# include <stdio.h>
void main()
{
    float f1(float, int), f2(int);
    float exp = 1.0, x; int n;
    printf("please input a number:");
    scanf("%f", &x); printf("%f", x);
    exp = exp + x;
    for(n = 2; n <= 19; n++) exp = exp + f1(x, n)/f2(n);
    printf("x = %f, exp = %8.4f\n", x, exp);
}
float f1(float x, int n)
{
}
float f2(int n)
{      }
```

(5) 用条件编译方法实现以下功能:从键盘输入一行电文,可以任意选择两种输出:一为原文输出;一为加密输出(将字母向后移动三位(如'a'变成'd',……,'z'变成'c'。其他字符不变)。用 `# define` 命令来控制是否要译成密码。