

实验名称 函数

学号 _____ 姓名 _____

一、实验目的

- 1.理解函数定义、函数声明和函数调用等不同概念。
- 2.熟悉函数调用时实参和形参的对应关系，理解“值传递”和的“地址传递”的不同特性。
3. 学会函数嵌套调用方法。
4. 掌握如何利用函数实现指定任务，加深对“模块化设计”思想的理解。
- 5.理解变量存储类的含义，熟悉局部变量和全局变量的概念和使用。

二、预习并回答问题

1. 以下正确的说法是 B。
建立函数的目的之一是：
A.提高程序执行效率
B.提高程序可读性
C.减少程序的篇幅
D.减少程序文件所占内存
2. 以下正确的说法是 B。
A.用户若需调用标准库函数，调用前必须重新定义
B.用户可以重新定义标准库函数，若如此，该函数将失去原有含义
C.系统根本不允许用户重新定义标准库函数
D.用户若需调用标准库函数，调用前不必使用#include 命令将该函数所在文件包含到用户源文件中，系统自动去调
- 3.C语言规定，函数返回值的类型是由 D 。
A.return 语句中的表达式类型所决定
B.调用该函数时的主调函数类型所决定
C.调用该函数时系统临时决定
D.在定义该函数时所指定的函数类型所决定

4. C语言规定，简单变量做实参时，它和对应形参之间的数据传递是__**B**__。
- A.地址传递 B.单向值传递
- C.由实参传给形参，再由形参传回给实参 D.由用户指定传递方式
- 5.若用数组名作为函数调用的实参，传递给形参的是__**A**__。
- A.数组的首地址
- B.数组第一个元素的值
- C.数组中全部元素的值
- D. 数组元素的个数
- 6.若使用一维数组名作函数实参，以下正确的说法是__**A**__。
- A.必须在主调函数中说明此数组的大小
- B.实参数组类型与形参数组类型可以不匹配
- C.形参数组的长度可以缺省说明，因此在被调函数中不需要考虑形参数组的大小
- D. 实参数组名与形参数组名必须一致
- 7.在C语言中，形参的默认存储类型是__**A**__。
- A.auto B.register C.static D. extern
- 8.以下正确的说法是__**A**__。
- 如果在一个函数的复合语句中定义了一个变量，则该变量
- A.只在该复合语句中有效 B.在该函数有效
- C.在本程序范围内有效 D.为非法变量

三、实验内容

1. 阅读下列程序并分析程序结果。

```
#include <stdio.h>
void main()
{ int a[3][3]={ 1,3,5,7,9,11,13,15,17};
  int sum;
  int func(int a[][3]);
  sum=func(a);
  printf("\nsum=%d\n",sum);
}
func(int a[][3])
{ int i,j,sum=0;
```

```

for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        { a[i][j]=i+j;
          if(i==j) sum+=a[i][j];
        }
return sum;
}

```

【分析】

sum=6

2. 阅读下列程序并分析程序结果。

```

#include <stdio.h>
float func( int a, int b)
{   return  a+b; }
void main( )
{   int a=2,b=3,c=8;
    printf(“%3.0f”,func((int)func(a+c,b),a-c));
}

```

【分析】

7

3. 阅读下列程序并分析程序结果。

```

#include <stdio.h>
func(int a,int b)
{   static int m=0,i=2;
    i+=m+1;
    m=i+a+b;
    return m;
}
main()
{   int k=4,m=1,p;
    p=func(k,m); printf(“%d,”,p);
    p=func(k,m);printf(“%d\n”,p);}

```

【分析】

8,17

4. 阅读下列程序并分析程序结果。

```

#include <stdio.h>
void num()
{   extern int x,y; int a=15,b=10;

```

```

    x=a-b;
    y=a+b;
}
int x,y;
main()
{ int a=7,b=5;
  x=a+b;
  y=a-b;
  num();
  printf(“%d,%d\n”,x,y);
}

```

【分析】

5,25

5.函数 del 的作用是删除有序数组 a 中的指定元素 x。已有调用语句 n=del(a,n,x); 其中实参 n 为删除前数组元素的个数,赋值号左边的 n 为删除后数组元素的个数。试填空并为之配备主调函数。

```

del(int a[],int n,int x)
{ int p,i;
  p=0;
  while(x>=a[p]&& p<n) ____ (1) ____ ;   for(i=p-1;i<n;i++) ____ (2) ____ ;       n=n-1;
  return n;
}

```

【主调函数】

(1) p=p+1

(2) a[i]=a[i+1]

6.以下程序计算 10 名学生某门课成绩的平均分。试填空并上机调试通过。

```

float average(float array[10])
{ int i; float aver,sum=array[0];
  for(i=1;____ (3) ____ ;i++) sum+=____ (4) ____ ;
  aver=sum/10;
  return aver;
}
main()
{ float score[10],aver; int i;
  printf( “\n input 10 scores:” );

```

```

for(i=0;i<10;i++) scanf( "%f" ,&score[i]);
aver= _____ (5) ;
printf( "\naverage score is %.2f\n" ,aver);
}

```

(3)i<10

(4)array[i]

(5)average(score)

7. 假定在一个整型数组 b 中，每一个元素都是不超过两位的正整数。编写程序，统计该数组全部元素中数字 0、1.....9 各出现多少次。并将统计结果输出出来。编程要求：（1）为整型数组赋予指定测试数据：static int b[]={5,26,74,56,1,27};（2）编写 void count(int a[],int b[],int k); 函数，其中，k 为 b 数组长度，a[10]中存放 b 数组中各位数字出现次数。

（3）由 main 函数输出统计结果。

【源程序】

```

#include <stdio.h>

void main()
{ static int b[]={5,26,74,26,1,27};
  int a[10]={0},i;
  void count(int a[],int b[],int k);
  count(a,b,6);
  for(i=0;i<10;i++) printf("%d:%d\n",i,a[i]);
}

void count(int a[],int b[],int k)
{ int i,num;
  for(i=0;i<k;i++)
  { num=b[i];
    while(num) a[num%10]++,num/=10;
  }
}

```

8. 输入 5×5 的整型数组，编写程序实现：

- （1）分别求出两条对角线上的各元素之和。（定义 diagonal_sum 函数）
- （2）求主对角线上行、列下标均为偶数的各元素之积。（定义 mul 函数）
- （3）求出该数组全部外侧元素之和。（定义 outside_sum 函数）

【源程序】

```

#include <stdio.h>

```

```

void diagonal_sum(int a[][5]);
void mul(int a[][5]);
void outside_sum(int a[][5]);
void main()
{ int a[5][5],i,j;
  for(i=0;i<5;i++)
    for(j=0;j<5;j++)
      scanf("%d",&a[i][j]);

  diagonal_sum(a);
  mul(a);
  outside_sum(a);
}
void diagonal_sum(int a[][5])
{ int sum1=0,sum2=0,i;
  for(i=0;i<5;i++)
    sum1+=a[i][i],sum2+=a[i][4-i];
  printf("main_diagonal=%d,assist_diagonal=%d\n",sum1,sum2);
}
void mul(int a[][5])
{ int multi=1,i;
  for(i=0;i<5;i++)
    if(i%2==0)multi*=a[i][i];
  printf("Multiply=%d\n",multi);
}
void outside_sum(int a[][5])
{ int i,j,sum1=0,sum2=0;
  for(i=0;i<5;i++)
    for(j=0;j<5;j++)
      sum1+=a[i][j];
  for(i=1;i<=3;i++)
    for(j=1;j<=3;j++)
      sum2+=a[i][j];
  printf("outside_sum=%d\n",sum1-sum2);
}

```

9. 主函数中输入若干整数，其值均在 1 至 100 的范围内，用-1 作为输入结束标志，试编写函数 f 用于统计每个整数出现的个数并交给主函数打印。

其中 f 的原型是：void f(int a[],int c[],int n);a 数组用以保存输入的若干整数，c 数组保存统计结果，n 为输入的整数个数。

例如：若输入的整数为 1 2 3 4 1 2 -1

则统计结果为 1: 2

2: 2

3: 1

4: 1

【源程序】

```
#include <stdio.h>
#define M 50
void f(int a[],int c[],int n);
void main()
{ int a[M],c[101]={0},i,n,num;
  n=0;
  printf("输入 1`100 间任一整数, -1 结束\n");
  scanf("%d",&num);
  while(num!=-1) {a[n++]=num; scanf("%d",&num);}
  f(a,c,n);
  printf("统计结果是: \n");
  for(i=1;i<=100;i++)  if(c[i]) printf("%d:%d\n",i,c[i]);
}
void f(int a[],int c[],int n)
{ int i;
  for(i=0;i<n;i++) c[a[i]]++;
}
```

10. 分别用不同技术编程求:

$sum=1!+2!+3!+\dots+10!$

①用非递归法求解 ②用静态局部变量法求解

【源程序 1】 非递归法

【源程序 2】 静态局部变量法