

实验 10 函数

一、实验目的

- (1) 掌握定义函数的方法;
- (2) 掌握函数实参与形参的对应关系以及“值传递”、“地址传递”的方式;
- (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。

二、实验内容和步骤

I.基础部分：理解函数和局部变量的概念

- (1) 给定程序 c10-1-1.c 的功能是:交换变量 m 和 n 的值。运行调试 c10-1-1.c, 观察变量 m 和 n 的值是否交换。

```
/* c10-1-1.c */
#include<stdio.h>
int main()
{
    int m,n,t;
    printf("please input m and n:");
    scanf("%d%d",&m,&n);
    t=m;m=n;n=t;
    printf("m=%d,n=%d\n",m,n);
    return 0;
}
```

- (2) 将 c10-1-1.c 中的 m 和 n 交换的功能通过函数实现，程序另存为 c10-1-2.c。

```
/* c10-1-2.c */
#include<stdio.h>
void swap(int m,int n)
{
    int t;
    t=m;m=n;n=t;
    printf("function swap:\n");
    printf("m=%d,n=%d\n",m,n);
}

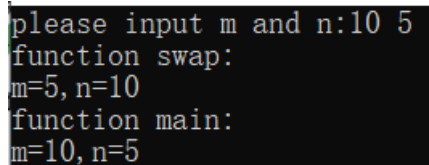
int main()
{
    int m,n;
    printf("please input m and n:");
```

```

scanf("%d%d",&m,&n);
swap(m,n);    //调用函数 swap()
printf("function main:\n");
printf("m=%d,n=%d\n",m,n);
return 0;
}

```

程序的运行结果如图 10-1 所示。



```

please input m and n:10 5
function swap:
m=5, n=10
function main:
m=10, n=5

```

图 10-1 程序 c10-1-2.c 的运行结果

分析并回答问题：

- ①函数 main() 和 swap() 中都有变量 m, 这 2 个 m 是同一个变量吗? 同理分析变量 n。
- ②为什么 swap() 函数中变量 m 和 n 的值交换了, 而 main() 函数中变量 m 和 n 的值没有交换?

(3) **修改**程序 c10-1-2.c, 另存为程序 c10-1-3.c。

```

/* c10-1-3.c */
#include<stdio.h>
void swap(int *m,int *n)
{
    int t;
    t=*m;*m=*n;*n=t;    /*数据交换*/
    printf("function swap:\n");
    printf("m=%d,n=%d\n",m,n);
}

int main()
{
    int m,n;
    printf("please input m and n:");
    scanf("%d%d",&m,&n);
    swap(&m,&n);          //调用函数 swap()
    printf("function main:\n");
    printf("m=%d,n=%d\n",m,n);
    return 0;
}

```

程序的运行结果如图 10-2 所示。

```

please input m and n:10 5
function swap:
m=1703724,n=1703720
function main:
m=5,n=10

```

图 10-2 程序 c10-1-3.c 的运行结果

分析下列问题：

- ① 为什么函数 swap() 中，变量 m、n 的值为 1703724、1703720？
- ② 为什么 main() 函数中，变量 m、n 的值实现了交换？

(4) 修改程序 c10-1-3.c，将语句：

```
t=*m;*m=*n;*n=t;
```

改为：

```
t=m;m=n;n=t;
```

程序另存为 c10-1-4.c。还能实现数据交换吗？为什么？

```

/* c10-1-4.c */
#include<stdio.h>
void swap(int *m,int *n)
{
    int t;
    t=m;m=n;n=t;          /******/
    printf("function swap:\n");
    printf("m=%d,n=%d\n",m,n);
}

int main()
{
    int m,n;
    printf("please input m and n:");
    scanf("%d%d",&m,&n);
    swap(&m,&n);    //调用函数 swap()
    printf("function main:\n");
    printf("m=%d,n=%d\n",m,n);
    return 0;
}

```

(5) 将程序改为通过全局变量实现，程序另存为 c10-1-5.c。这个程序可以实现变量 m、n 值的交换吗？为什么？

```

/* c10-1-5.c */
#include<stdio.h>
int m,n;                //全局变量
void swap()
{
    int t;

```

```

        t=m;m=n;n=t;
        printf("function swap:\n");
        printf("m=%d,n=%d\n",m,n);
    }

    int main()
    {
        printf("please input m and n:");
        scanf("%d%d",&m,&n);
        swap( );           //调用函数 swap()
        printf("function main:\n");
        printf("m=%d,n=%d\n",m,n);
        return 0;
    }

```

(6) 在程序 c10-1-5.c 中，在 main() 函数中增加声明语句：

```

        int m,n;
程序另存为 c10-1-6.c。
/*c10-1-6.c*/
#include<stdio.h>
int m,n;
void swap()
{
    int t;
    t=m;m=n;n=t;
    printf("function swap:\n");
    printf("m=%d,n=%d\n",m,n);
}

int main()
{
    int m,n;           /*****/
    printf("please input m and n:");
    scanf("%d%d",&m,&n);
    swap( );           //调用函数 swap()
    printf("function main:\n");
    printf("m=%d,n=%d\n",m,n);
    return 0;
}

```

运行结果如图 10-3 所示。

```

please input m and n:10 5
function swap:
m=0, n=0
function main:
m=10, n=5

```

图 10-3 程序 c10-1-6.c 的运行结果

分析下列问题：

- ① 这个程序为什么不能实现变量 m、n 值的交换？
- ② swap() 函数中输出 m 和 n 的值为什么是 0？

II. 提高部分

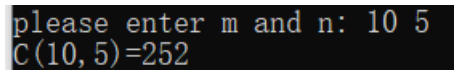
1、程序填空

说明：程序有多个空 (1)、(2) …… 需要补充完整。请将程序中的 __ (1) __、__ (2) __ …… 删除后，在相应的位置填入正确答案并调试直到得到正确结果为止。

注意：不要随意改动程序，不得增行或删行，也不得更改程序的结构！

(1) 给定程序 c10-2-1.c 的功能是：计算 $C_m^n = \frac{m!}{n!(m-n)!}$ 的值。程序运行结果如图 10-4

所示，请把程序补充完整。



```
please enter m and n: 10 5
C(10, 5)=252
```

图 10-4 程序 c10-2-1.c 的运行结果

```
/*    c10-2-1.c    */
#include <stdio.h>
/*****found*****/
int  fac(__(1)__)    /*定义求阶乘函数 jf*/
{
    int  i;
    int  t=1;
    for(i=1; i<=n; i++)
        t=t*i;
/*****found*****/
    ____ (2) ____;
}

int  main( )
{
    int m,n,t,cmn;
    printf("please enter m and n: ");
    scanf("%d%d", &m, &n);
    if(m<n)
    {    t=m;m=n;n=t;    }
    if(m<0||n<0)
    {    printf("input error!"); return 0;    }
/*****found*****/
    cmn=__(3)____;    /*调用 fac()函数*/
}
```

```

        printf("C(%d,%d)=%d\n",m,n,cmn) ;
        return 0;
    }

```

(2) 给定程序 c10-2-2.c 的功能是：求两个正整数 m, n 之间的非素数之和（包含 m 和 n）。程序运行结果如图 10-5 所示，请把程序补充完整。

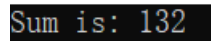


图 10-5 程序 c10-2-2.c 的运行结果

```

/*  c10-2-2.c  */
#include <math.h>
#include <stdio.h>

/*****found*****/
_____(1)_____
int main()
{
    int m=2,n=20,i,s;
    s = 0;
    for ( i=m; i<=n; i++)
/*****found*****/
        if( ____ (2) ____ ) s += i;
    printf("Sum is: %d\n",s);
    return 0;
}

/*****found*****/
int  prime( __ (3) __ )
{
    int k,j;
    k=sqrt(m);
    for (j=2;j <= k; j++)
        if (m % j == 0) break;
    if( j<=k )
        return 0;
    else
        return 1 ;
}

```

(3) 给定程序 c10-2-3.c 的功能是：求一维数组 a 中既不能被 3 整除也不能被 5 整除的元素之和。程序运行结果如图 10-6 所示，请把程序补充完整。

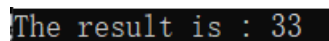
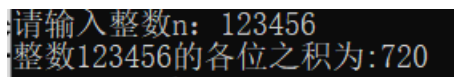


图 10-6 程序 c10-2-3.c 的运行结果

```
/* c10-2-3.c */
#include <stdio.h>
int fun(int arr[], int n );
int main()
{
    int a[12]={ 1,8,5,10,6,18,7,2,4,12,9,11};
    /******found*****/
    printf("The result is : %ld\n",__(1));
    return 0;
}
/******found*****/
int fun(__(2))
{
    int i;
    int s=0;
    for (i=0;i<n;i++)
        if ( arr[i]%3!=0&&arr[i]%5!=0 )
            /******found*****/
            __ (3) __;
    return (s);
}
```

(4)给定程序 c10-2-4.c 的功能是：求正整数 n 的各位之积。程序运行结果如图 10-7 所示，请把程序补充完整。



```
请输入整数n: 123456
整数123456的各位之积为:720
```

图 10-7 程序 c10-2-4.c 的运行结果

```
/* c10-2-4.c */
#include <stdio.h>
/******found*****/
__(1)__func(long m)
{
    long k=1;
    do
    {
        k=k*(m%10);
        /******found*****/
        ____(2)____
        /******found*****/
    }while(__(3));
    return(k);
}
```

```

}
int main( )
{
    long n;
    printf("请输入整数 n: ");
    scanf("%ld",&n);
    printf("整数%ld 的各位之积为:%ld\n",n,func(n));
}

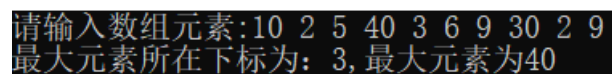
```

2、程序改错

说明：程序中有错误，错误都在提示行： `/******found******/` 的下面一行。请改正程序中的错误，编译运行使它能得出正确的结果。

注意：程序中的其它地方不要随意改动，不得增行或删除行，也不得更改程序的结构！

(1) 给定程序 `c10-2-5.c` 的功能是：调用 `findmax` 函数求数组中值最大的元素在数组中的下标，函数 `findmax` 返回最大元素所在下标。程序的运行结果如图 10-8 所示，请改正程序中的错误，并运行出正确的结果。



```

请输入数组元素:10 2 5 40 3 6 9 30 2 9
最大元素所在下标为: 3, 最大元素为40

```

图 10-8 程序 `c10-2-5.c` 的运行结果示例

```

/*  c10-2-5.c  */
#include <stdio.h>
/******found******/
int findmax( int s[],n)
{
    /******found******/
    int j,p,n,s[n];
    p=0;
    for (j=1;j<n;j++)
    {
        if (s[j]>s[p])  p=j;
    }
    /******found******/
    return  s[p];
}
int  main( )
{
    int a[10],i,k;
    printf("请输入数组元素:");
    for (i=0;i<10;i++)
        scanf("%d",&a[i]);
    k=findmax(a,10);
    printf("最大元素所在下标为: %d,最大元素为%d\n",k,a[k]);
}

```



```

    return 0;
}

```

(2) 给定程序 c10-2-6.c 的功能是：求一维数组 a 中值为偶数的元素之和。程序的运行结果如图 10-9 所示，请改正程序中的错误，并运行出正确的结果。

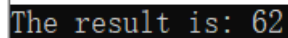


图 10-9 程序 c10-2-6.c 的运行结果示例

```

/*  c10-2-6.c  */
#include <stdio.h>
/*****found*****/
void sum ( int  arr[ ], int n )
{
    int  i,s;
    s = 0;
    for ( i=0; i<n; i++)
        /*****found*****/
        if (arr[i] % 2 == 0)  s = s + i;
    return (s);
}

int main()
{
    int a[10]={ 10,4,2,7,3,12,5,34,5,9},s;
    /*****found*****/
    s=sum( a[10] );
    printf("The result is: %d\n", s);
    return 0;
}

```

(3) 给定程序 c10-2-7.c 的功能是：用二分法在一个有序数组中查找给定值。如果找到，函数 bin_search()返回找到元素所在的下标，如果没找到，函数返回-1。程序的运行结果如图 10-10 所示，请改正程序中的错误，并运行出正确的结果。

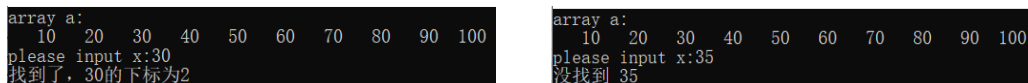


图 10-10 程序 c10-2-7.c 的运行结果示例

```

/*  c10-2-7.c  */
#include<stdio.h>
#define N 10

int bin_search(int arr[],int x,int low,int high)

```

//low 和 high 分别为待查找的数组的下标的最小值和最大值，x 为待查找的元素

```
{
    int mid;
    /*******found*****/
    while(high<=low)
    {
        mid=(low+high)/2;
        if(x==arr[mid]) return mid;
        else
            /*******found*****/
            if(x<arr[mid]) low=mid+1;
            else
                high=mid-1;
    }
    return -1;
}

int main()
{
    int a[N]={ 10,20,30,40,50,60,70,80,90,100};
    int x,find_x,i;
    printf("array a:\n");
    for(i=0;i<N;i++)
        printf("%5d",a[i]);
    printf("\n");
    printf("please input x:");
    scanf("%d",&x);
    /*******found*****/
    find_x=bin_search(a[],x,1,N);
    if(find_x==-1)
        printf("没找到 %d\n",x);
    else
        printf("找到了， %d 的下标为%d\n",x,find_x);

    return 0;
}
```

3、程序设计

说明：下列程序均需要编写对应的 main() 函数，调试运行得到结果。

(1) 编写程序 c10-2-8.c，函数实现的功能是：求两个整数的最大公约数，并根据求得的最大公约数计算最小公倍数。

求最大公约数的函数原型为：

```
int GCD(int m,int n);
```

函数返回 m 和 n 的最大公约数。

求最小公倍数的函数原型为：

```
int LCM(int m,int n);
```

函数返回 m 和 n 的最小公倍数。

(2) 编写程序 **c10-2-9.c**, 函数实现的功能是: 将一个十进制整数转换为对应的二进制数, 十进制数从键盘输入。

转换函数原型为:

```
int convert(int m, int binary[]);
```

其中, m 为待转化为二进制的十进制数, 转化后的结果保存在数组 binary[] 中, 函数返回二进制数的长度。

编程提示:

➤ 以十进制数 18 为例, 求二进制过程如下:



➤ 每求得一位二进制, 将该二进制保存到数组 binary[] 中, 因此, 数组 binary[] 中的二进制是逆序存放的, 二进制的位数 (长度) 由函数返回。因此, 十进制数 18 转化后, 数组 binary[] 中的数据如下:

下标	0	1	2	3	4	5	6	7	8	9
二进制	0	1	0	0	1						

思考: 如果将十进制数转换为十六进制数, 应对程序的哪个语句进行修改? 怎样修改?

(3) 编写程序 **c10-2-10.c**, 函数实现的功能是: 将一个字符串的奇数位 (第 1, 3, 5, 7, 9, 位) 字符复制到另一个字符串中。

例如: 当字符串为 "This Is a c Program", 则得到的新字符串为 "Ti sacPorm"。

函数原型为:

```
void fun(char str1[],char str2[]);
```

其中, 数组 str1[] 是原字符串, 数组 str2[] 是新字符串。

(4) 编写程序 **c10-2-11.c**, 实现的功能是: 通过函数实现选择法排序。

选择法排序的函数原型为:

```
void select_sort(int arr[],int n);
```

其中, 数组 arr[] 为待排序的数组, n 为数组的大小。

(5) 编写程序 **c10-2-12.c**, 函数实现的功能是: 给定的一个 N×N 的二维数组, 实现数组转置。

转置函数原型如下:

```
void transposition(int x[N][N]);
```

(6) 编写程序 **c10-2-12.c**, 函数实现的功能是: 用递归法求 n 阶勒让德多项式, 递归公式为:

$$P_n(x) = \begin{cases} 1 & (n=0) \\ x & (n=1) \\ ((2n-1)*x*P_{n-1}(x)-(n-1)*P_{n-2}(x))/n & (n>1) \end{cases}$$

III.拓展部分：熟悉模块化程序设计

(1) 学生成绩管理系统 (V3.0 版)

文件 `score.txt` 中存放若干同学的学号及高数、英语、C 语言 3 门课的成绩，格式如下：

```
1001  90  80  70
1002  85  78  80
1003  60  70  76
.....
```

编写程序 `c10-3-1.c`，实现如下功能：

- ① 从 `score.txt` 中逐个读入每个同学的成绩，用一个函数实现；
- ② 计算每个同学平均分，用一个函数实现；
- ③ 成绩按平均分排序，用一个函数实现；
- ④ 屏幕输出结果并写入文件 `score_avg.txt` 中，用一个函数实现。

文件 `score.txt` 中的记录应该为：

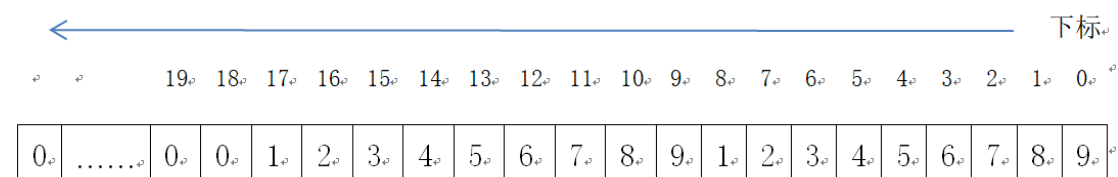
学号	高数	英语	C 语言	平均分
1002	85	78	80	81.00
1001	90	80	70	80.00
1003	60	70	76	68.67

.....

(2) 挑战类型表示的极限——大数存储和运算 (V2.0版)

C 语言中无论是 `int` 还是 `long int` 类型，数据的存储范围都是有限的，常用的整型数据在内存中占用四个字节，能表示的数据范围是 $-2147483648 \sim 2147483647$ ，即使在 C++ 中，最大的 `long long` 数据类型也只能存储 $-9223372036854775808 \sim 9223372036854775807$ 范围内的数据，当整数的值超过这个存储范围，将不能被正确表示。

一种解决方法是用一个 `int` 类型数组存放和表示大数，一个数组元素存放大数中的一位。例如：定义一个 `int A[200]`，可以存放一个最多为 200 位的无符号整数。假设有一个大整数 123456789123456789，保存到 A 数组中，数组中的数据表示如下：



编写程序 `c10-3-2.c` 实现如下功能：

- ① 通过函数实现大整数输入，输入函数的原型为：

```
void input(int bigint[]);
```

从键盘输入的数据存储在 `bigint` 数组中。输入函数中需要判断输入数据是否合法：（a）不能输入非数值字符，如果输入了非法字符，结束程序运行；（b）输入的位数不能超过定义的大整数的最大位数。

提示：可以按字符串的形式输入数据，然后将字符串转化为数值。

- ② 通过函数实现大整数输出，要求可以输出任意数值（包括 0）的无符号大整数。函数原型如下：

```
void output(int bigint[]);
```

其中，参数 `bigint[]` 是需要输出的大整数。

③ 通过函数实现大整数的加法，函数原型为：

```
void add(int bigint1[], int bigint2[], int bigint3[]);
```

其中，参数bigint1和bigint2为需要相加的2个大整数，相加结果保存到bigint3中。

提示：相加过程中要考虑进位。

④ 实现2个大整数相比较，函数原型为：

```
int cmp_bigint(int bigint1[], int bigint2[]);
```

其中，参数bigint1和bigint2为待比较的2个大数。如果bigint1>bigint2，函数返回1；如果bigint1==bigint2，函数返回0；如果bigint1<bigint2，函数返回-1。

⑤ 实现两个大整数的减法运算，函数原型为：

```
void sub(int bigint1[], int bigint2[], int bigint3[]);
```

函数将大整数 bigint1 减去 bigint2，结果保存到 bigint3 中。为了不出现负数，要保证被减数大于减数。

提示：相减时要考虑借位问题。

⑥ 实现2个大整数bigint1和bigint2相乘的功能，结果保存在bigint3中。函数原型为：

```
void multiply(int bigint1[], int bigint2[], int bigint3[]);
```

提示：两个大数相乘的一种算法思路：

例如：bigint1=12345，bigint2=67890

计算过程如下：

第一步：计算各位相乘的累加和，不考虑进位。										
					1	2	3	4	5	
					×)	6	7	8	9	0
<hr/>										
					+) 0	0	0	0	0	0
					9	18	27	36	45	
				8	16	24	32	40		
			7	14	21	28	35			
		6	12	18	24	30				
<hr/>										
		6	19	40	70	100	94	76	45	0
<hr/>										
未进位时，bigint3 中的数据如下：										
Bigint3:										
0	6	19	40	70	100	94	76	45	0
		8	7	6	5	4	3	2	1	0
第二步：从低位到高位计算处理 bigint3 中的进位。										
进位后的 bigint3 中数据如下：										
Bigint3:										
0	8	3	8	1	0	2	0	5	0
		8	7	6	5	4	3	2	1	0

⑦ 设计一个菜单，实现上述功能的调用。