

实验 3 函数与程序结构实验

3.1 实验目的

- (1) 熟悉和掌握函数的定义、声明；函数调用与参数传递方法；以及函数返回值类型的定义和返回值使用。
- (2) 熟悉和掌握不同存储类型变量的使用。
- (3) 熟悉多文件编译技术。

3.2、实验内容

1. 源程序改错题

下面是计算 $s=1!+2!+3!+\dots+n!$ 的源程序，在这个源程序中存在若干语法和逻辑错误。要求在计算机上对这个例子程序进行调试修改，使之能够正确完成指定任务。

```
1  #include "stdio.h"
2  void main(void)
3  {
4      int k;
5      for(k=1;k<6;k++)
6          printf("k=%d\tthe sum is %ld\n",k,sum_fac(k));
7  }
8  long sum_fac(int n)
9  {
10     long s=0;
11     int i;
12     long fac;
13     for(i=1;i<=n;i++)
14         fac*=i;
15     s+=fac;
16     return s;
17 }
```

解答：

- (1) 错误修改：
 - 1) 第1行，没有声明函数sum_fac()，正确形式为：
long sum_fac(int n);
 - 2) 第15行，s+=fac应放入for循环内，正确形式为：
for (i = 1; i <= n; i++)

```

    {
        fac *= i;
        s += fac;
    }

```

3) 第12行, fac未初始化, 正确的形式为:

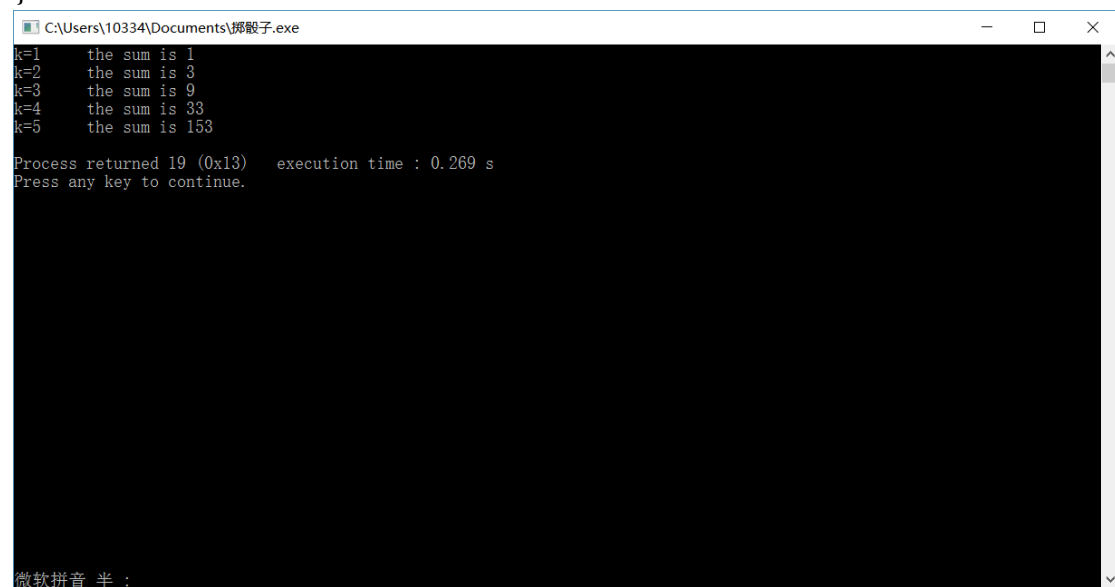
```
Long fac=1;
```

(2) 错误修改后运行结果:

```

#include<stdio.h>
long sum_fac(int n);
void main(void)
{
    int k;
    for (k = 1; k<6; k++)
        printf("k=%d\tthe sum is %ld\n", k, sum_fac(k));
}
long sum_fac(int n)
{
    long s = 0;
    int i;
    long fac=1;
    for (i = 1; i <= n; i++)
    {
        fac *= i;
        s += fac;
    }
    return s;
}

```



```

C:\Users\10334\Documents\骰骰子.exe
k=1    the sum is 1
k=2    the sum is 3
k=3    the sum is 9
k=4    the sum is 33
k=5    the sum is 153

Process returned 19 (0x13)   execution time : 0.269 s
Press any key to continue.

```

2. 源程序修改替换题

(1) 修改第 1 题中 sum_fac 函数，使其计算量最小。

(2) 修改第 1 题中 sum_fac 函数，计算 $s = 1 + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!}$ 。

解答：

(1) 替换后的程序如下所示：

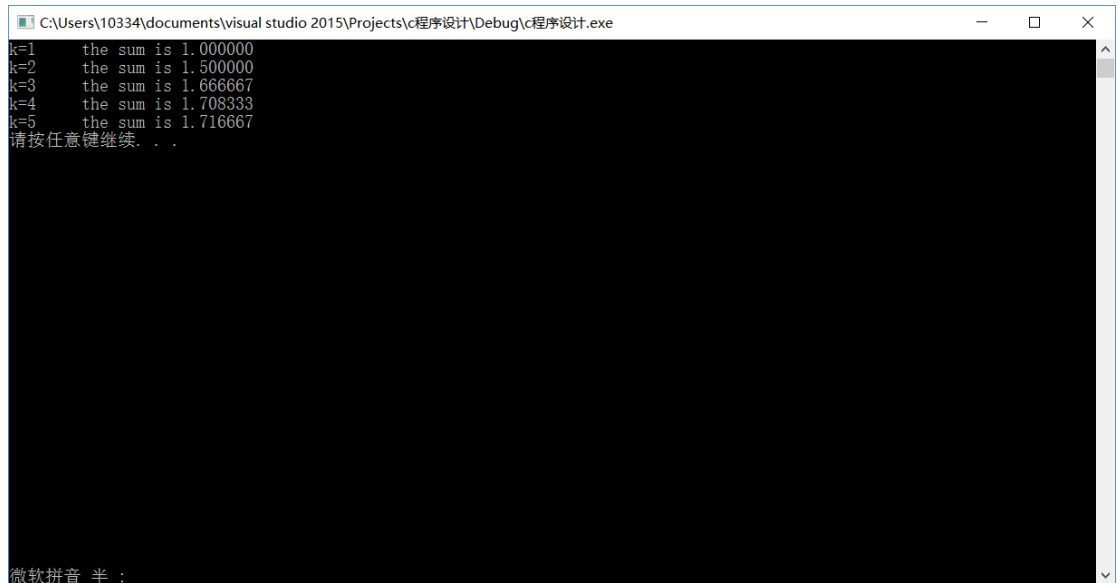
```
long sum_fac(int n)
{
    static long fac=1;
    fac *= n;//计算阶乘
    static long sum = 0;//存储阶乘和
    return sum+=fac;
}
```

(2) 替换后的程序如下所示：

```
#include<stdio.h>
double sum_fac(int n);
void main(void)
{
    int k;
    for (k = 1; k<6; k++)
        printf("k=%d\tthe sum is %lf\n", k, sum_fac(k));
    system("pause");
}

double sum_fac(int n)
{
    static long fac = 1;
    fac *= n;//计算阶乘
    static double sum = 0;//存储阶乘和
    return sum += 1.0/fac;
}
```

} 运行结果如下：



```
C:\Users\10334\documents\visual studio 2015\Projects\c程序设计\Debug\c程序设计.exe
k=1    the sum is 1.000000
k=2    the sum is 1.500000
k=3    the sum is 1.666667
k=4    the sum is 1.708333
k=5    the sum is 1.716667
请按任意键继续...
```

3. 跟踪调试题

计算 fibonacci 数列前 n 项和的程序如下：

其中，`long sum=0,*p=∑`声明 p 为长整型指针并用 `&sum` 取出 sum 的地址

对 p 初始化。`*p` 表示引用 p 所指的变量（`*p` 即 `sum`）。

```
void main(void)
{
    int i,k;
    long sum=0,*p=&sum;
    scanf("%d",&k);
    for(i=1;i<=k;i++){
        sum+=fibonacci(i);
        printf("i=%d\tthe sum is %ld\n",i,*p);
    }
}

long fibonacci(int n)
{
    if(n==1 || n==2)
        return 1;
    else
        return fibonacci(n-1)+fibonacci(n-2);
}
```

单步执行程序，观察 p,i,sum,n 值。

（1）刚执行完 `scanf("%d",&k);`语句，p,i 值是多少？

解答：

P= 0x012ffc2c

I= -858993460

(2) 从 fibonacci 函数返回后光条停留在哪个语句上？

解答：

sum+=fibonacci(i);

(2) 进入 fibonacci 函数，watch 窗口显示的是什么？

解答：

自动窗口		
名称	值	类型
返回 fibonacci	1	long
&k	0x012ffc38 (5)	int *
&sum	0x012ffc2c (2)	long *
*p	2	long
i	1	int
k	5	int
sum	2	long

(4) 当 i=3，从调用 fibonacci 函数到返回，n 值如何变化？

解答：

3 变为 11604038

4. 编程设计题

(1) 编程让用户输入两个整数，计算两个数的最大公约数并且输出之（要求用递归函数实现求最大公约数）。同时以单步方式执行该程序，观察递归过程。

解答：

1) 解题思路：

运用辗转相除法，输入两个数 x，y，通过计较大小保证 y>x,作为参数传递给 max_commom_divisor()函数，函数模拟辗转相除法计算最大公约数。

2) 程序清单

```
#include<stdio.h>
int max_common_divisor(int y, int x);
int main(void)
{
    int x, y,temp;
    scanf("%d %d", &x, &y);
    while (x)
    {
        if (x > y)
        {
            temp = x;
            x = y;
```

```

        y = temp;
    }//保证y>=x
    printf("%d\n", max_common_divisor(y,x));
    scanf("%d %d", &x, &y);
}
}
int max_common_divisor(int y, int x)
{
    int m;
    m = y%x;
    if (!m)
        return x;
    else return max_common_divisor(x, m);
}

```

3) 测试

a) 测试数据：

1 6

36 99

15 6

50 100

0 0（退出）

b) 测试结果：

```

C:\Users\10334\Documents\税金.exe
1 6
1
36 99
9
15 6
3
50 100
50
0 0
Process returned 0 (0x0)   execution time : 17.197 s
Press any key to continue.

```

微软拼音 半：

(2) 编程验证歌德巴赫猜想：一个大于等于 4 的偶数都是两个素数之和。

编写一个程序证明对于在符号常量 BEGIN 和 END 之间的偶数这一猜测成立。例如，如果 BEGIN 为 10，END 为 20，程序的输出应为：

GOLDBACH'S CONJECTURE:

Every even number $n \geq 4$ is the sum of two primes.

10=3+7

12=5+7

.....

20=3+17

解答：

1) 解题思路：

Isprime()函数判断素数，Goldbach_num()验证哥德巴赫猜想，输入 begin 和 end，while 中判断 begin 是否为奇数，如果为奇数将其加 1，i 从 begin 到 end 依次加 2，且每一次判断 i 由两个素数之和，Goldbach_num()函数从 2 遍历到参数 x 寻找素数，当找到素数 i 时判断 x-i 是否为素数，若为，则输出，否则继续寻找下一个。

2) 程序清单

```
#include<stdio.h>
#include<math.h>
int IsPrime(int n);
int Goldbach_num(int x);
int main(void)
{
    int begin, end, i;
    scanf("%d %d", &begin, &end);
    while (begin)
    {
        if (begin % 2)
            begin++;
        for (i = begin; i <= end; i += 2)
        {
            Goldbach_num(i);
        }
        putchar('\n');
        scanf("%d %d", &begin, &end);
    }
}

int IsPrime(int n)
```

```

{
    int s, i;

    for (i = 2; i <= sqrt(n); i++)
    {
        s = n%i;
        if (s == 0)
            return 0;
    }
    return 1;
}
int Goldbach_num(int x)
{
    int j, k;
    for (j = 2; j <= x; j++)
    {
        if (IsPrime(j) && IsPrime(x - j))
        {
            printf("%d=%d+%d\n", x, j, x - j);
            j = x + 1;
        }
    }
}
}

```

3) 测试

a) 测试数据:

4 10

5 9

10 23

b) 测试结果:



```
C:\Users\10334\documents\visual studio 2015\Projects\c程序设计\Debug\c程序设计.exe
4 10
4=2+2
6=3+3
8=3+5
10=3+7

5 9
6=3+3
9=3+5

10 23
10=3+7
12=5+7
14=3+11
16=3+13
18=5+13
20=3+17
22=3+19
```

5. 选做 s 题

1、设 file1.c 如下：

```
#include <stdio.h>
int x,y; /* 外部变量的定义性说明 */
char ch; /* 外部变量的定义性说明 */
void main(void)
{
    x=10;
    y=20;
    ch=getchar();
    printf("in file1 x=%d,y=%d,ch is %c\n",x,y,ch);
    func1();
}
```

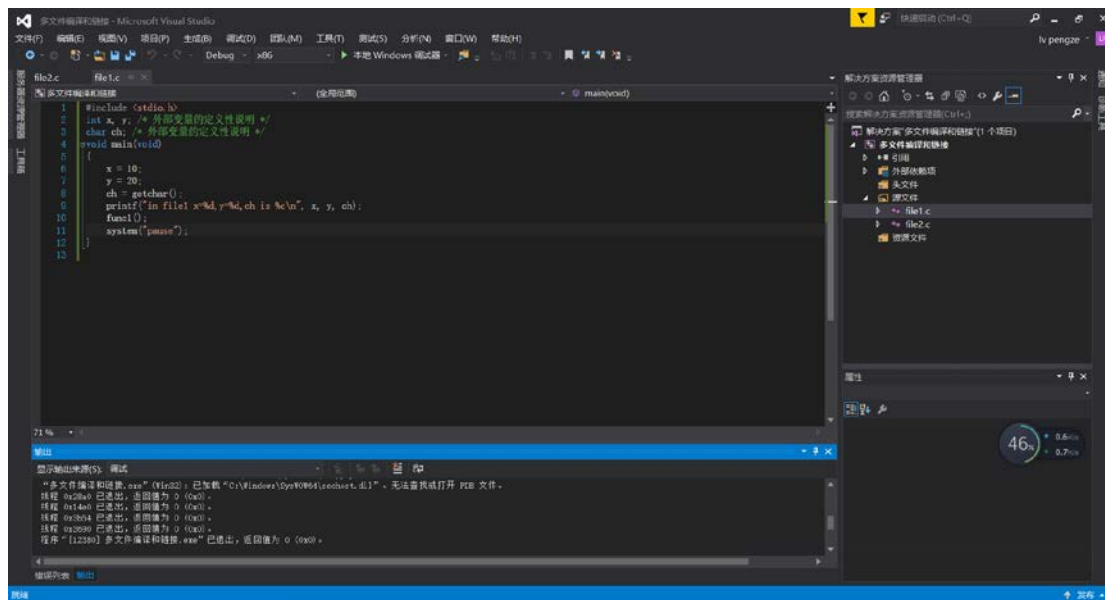
file2.c 如下：

```
extern int x,y; /* 外部变量的引用性说明 */
extern char ch; /* 外部变量的引用性说明 */
void func1(void)
{
    x++;
    y++;
    ch++;
    printf("in file2 x=%d,y=%d,ch is %c\n",x,y,ch);
}
```

试用 TCC 进行多文件编译和链接。然后在 DOS 环境下运行生成的可执行文件。

```
c:\users\10334\documents\visual studio 2015\Projects\多文件编译和链接\Debug\多文件编译和链接.exe
6
in file1 x=10,y=20,ch is 6
in file2 x=11,y=21,ch is 7
请按任意键继续. . .
```

微软拼音 半：



3.3、实验总结：

通过本次实验，我掌握了分部调试，通过分步调试，我可以清楚的看到各变量的变化，有助于我充分理解程序以及修改程序错误。

其次，我掌握了静态变量的用法和多文件链接和编译。

验证哥德巴赫猜想这一题，可以看出计算机与数学问题的联系，可以通过计算机帮助人们解决许多难题。

此外，在编写程序时，一定要十分小心，尤其在循环语句的编写中，要注意循环继续和终止的条件。如果出现错误，要仔细的调试，一步一步的来，不能急躁。在编写程序时，尽量使程序的计算量要小，这需要掌握好变量的种类运用。通过这次实验，我对程序调试的熟练度大大增加，了解了多文件程序的编译与链接。并且了解一些新的算法思想。这使我的思维角度大大拓宽。