

实验 2 流程控制实验

2.1、实验目的

- (1) 掌握复合语句、if 语句、switch 语句的使用，熟练掌握 for、while、do-while 三种基本的循环控制语句的使用，掌握重复循环技术，了解转移语句与标号语句。
- (2) 练习循环结构 for、while、do-while 语句的使用。
- (3) 练习转移语句和标号语句的使用。
- (4) 使用 Turbo C 2.0 集成开发环境中的调试功能：单步执行、设置断点、观察变量值。

2.2、实验内容

1. 源程序改错题

下面是计算 $s=n!$ 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。例如， $8! = 40320$ 。

程序清单：

```
#include <stdio.h>
void main(void)
{
    int i, n, s = 1;
    printf("Please enter n:");
    scanf("%d", &n); // 键盘读取整型n应加上&
    for (i = 1; i <= n; i++) // for() 各参数应用分号分开，而不是逗号
        s = s*i;
    printf("%d! = %d", n, s);
}
```

解答：

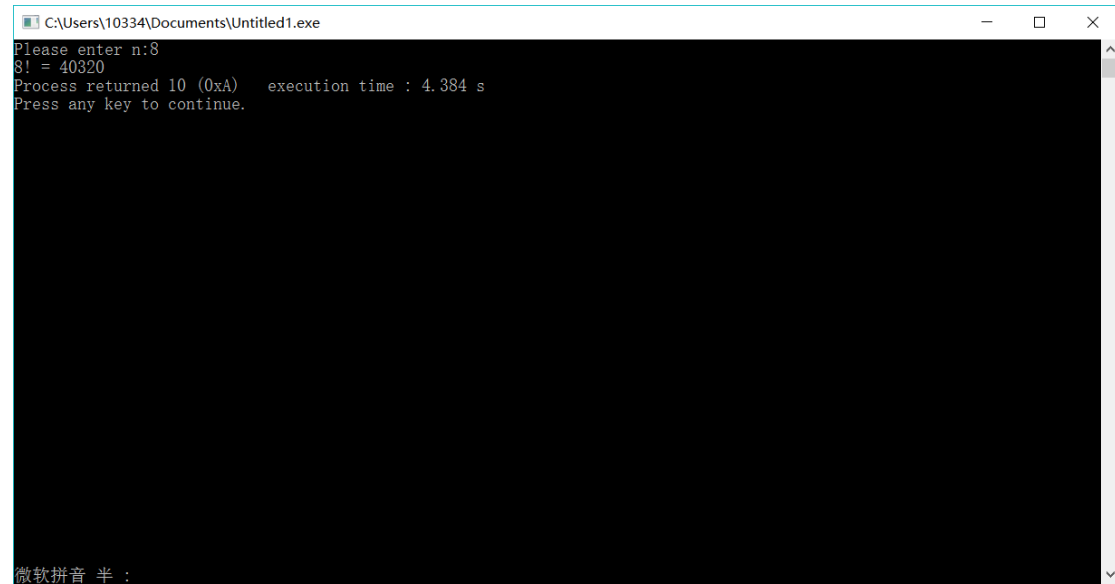
程序清单：

```
#include <stdio.h>
void main(void)
{
    int i, n, s = 1;
    printf("Please enter n:");
    scanf("%d", &n); // 键盘读取整型n应加上&
    for (i = 1; i <= n; i++) // for() 各参数应用分号分开，而不是逗号
```

```
        s = s*i;
    printf("%d! = %d", n, s);
}
```

测试样例：8

测试结果：



2. 源程序修改替换题

(1) 修改第 1 题，分别用 while 和 do-while 语句替换 for 语句。

/*while语句写法*/

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

```
void main(void)
```

```
{
```

```
    int i=1, n, s = 1;
```

```
    printf("Please enter n:");
```

```
    scanf("%d", &n);
```

```
    while (i <= n)
```

```
    {
```

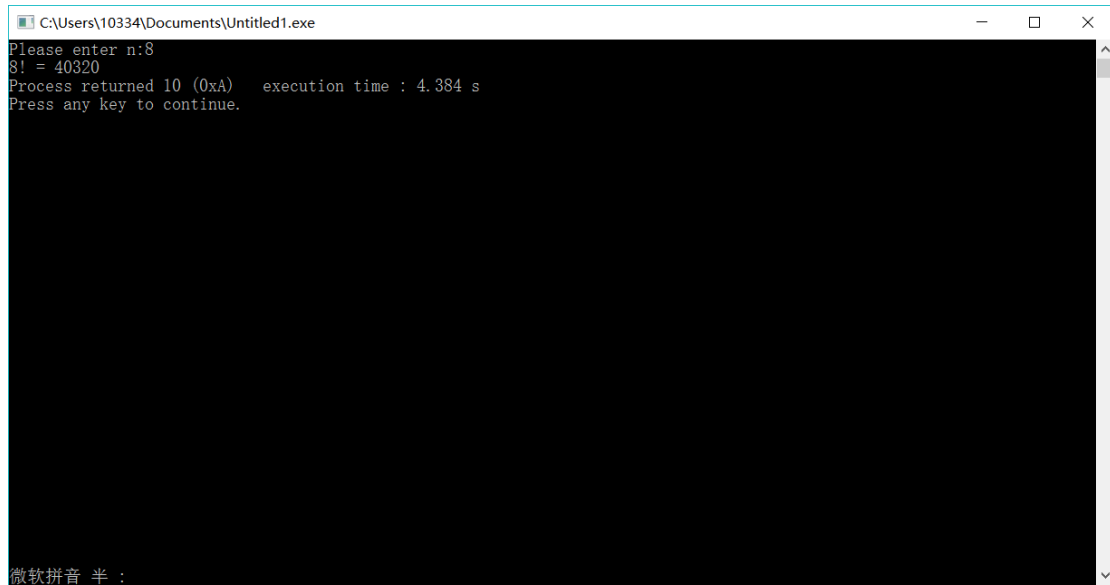
```
        s = s*i;
```

```
        i++;
```

```
    }
```

```
    printf("%d! = %d", n, s);
```

```
}
```



微软拼音 半 :

/*do-while语句写法*/

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

```
void main(void)
```

```
{
```

```
    int i=1, n, s = 1;
```

```
    printf("Please enter n:");
```

```
    scanf("%d", &n);
```

```
    do
```

```
    {
```

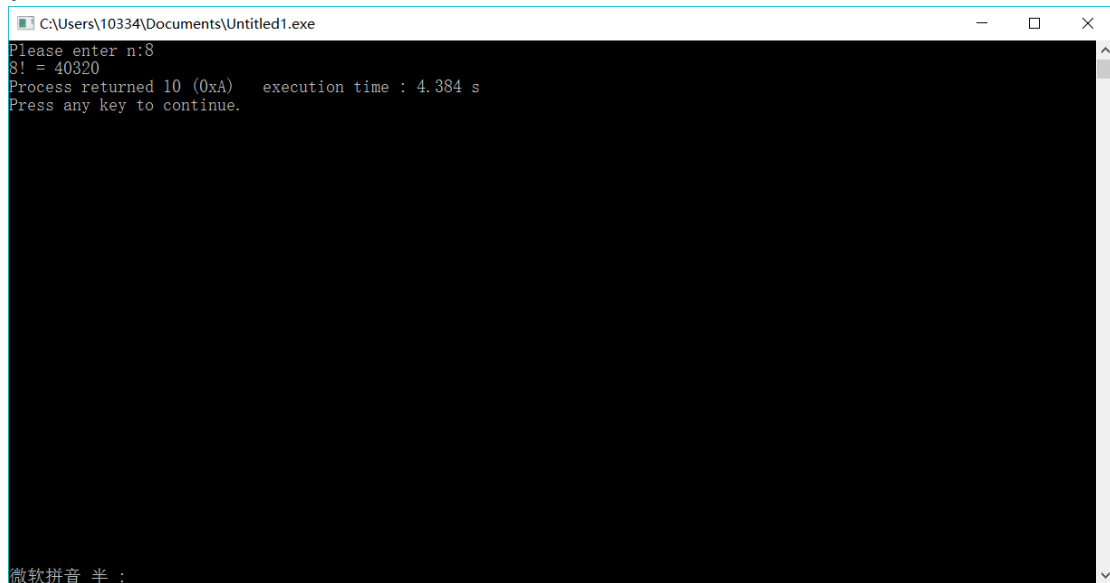
```
        s = s*i;
```

```
        i++;
```

```
    } while (i <= n);
```

```
    printf("%d! = %d", n, s);
```

```
}
```



(2) 修改第 1 题，输入改为“整数 S”，输出改为“满足 $n! \geq S$ 的最小整数 n”。例

如输入整数 40310，输出结果为 n=8。

/*do-while语句写法*/

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

```
int divisible_max(int S);
```

```
void main(void)
```

```
{
```

```
    int S, result;
```

```
    printf("Please enter S:");
```

```
    scanf("%d", &S);
```

```
    result = divisible_max(S);
```

```
    printf("%d", result);
```

```
}
```

```
int divisible_max(int S)
```

```
{
```

```
    int i = 1, s = 1;
```

```
    do
```

```
    {
```

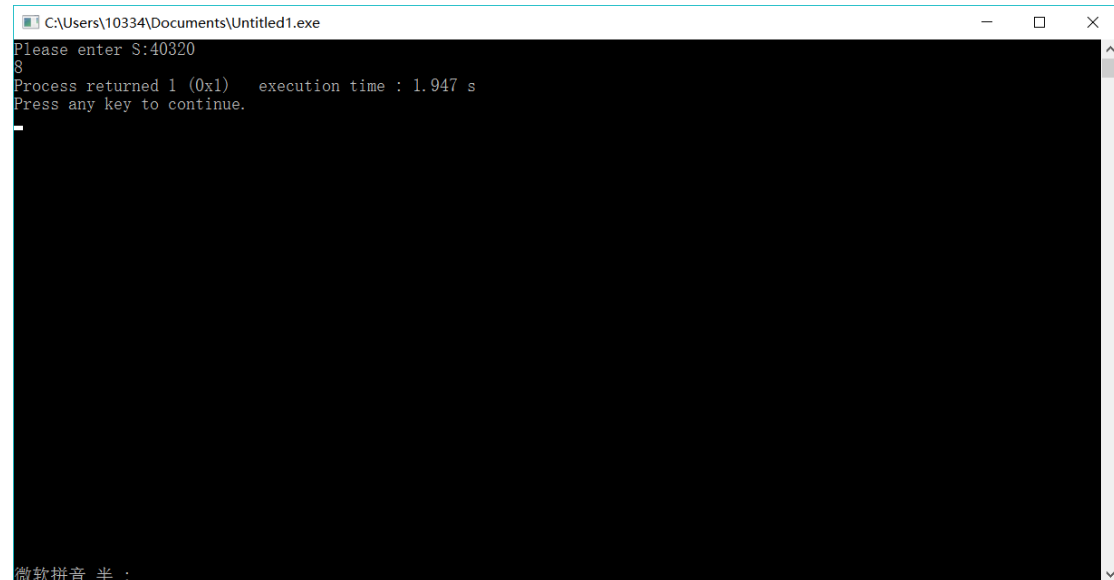
```
        s *= i;
```

```
        i++;
```

```
    } while (s < S);
```

```
    return i - 1;
```

```
}
```



3. 编程设计题

(1) 打印如下杨辉三角形。

```
    1
   1 1
```

/*第 0 行 */

/*第 1 行 */

```

          1   2   1                               /*第 2 行 */
        1   3   3   1
      1   4   6   4   1
    1   5   10  10  5   1
  1   6   15  20  15  6   1
1   7   21  35  35  21  7   1
1   8   28  56  70  56  28  8   1
1   9   36  84 126 126 84  36  9   1

```

每个数据值可以由组合 C_i^j 计算 (表示第 i 行第 j 列位置的值), 而 C_i^j 的计算如下:

$$C_i^0 = 1 \quad (i=0,1,2,\dots)$$

$$C_i^j = C_i^{j-1} * (i - j + 1) / j \quad (j=0,1,2,3,\dots,i)$$

本程序中为了打印出金字塔效果, 要注意空格的数目。一位数之间是 3 个空格, 两位数之间有 2 个空格, 3 位数之间只有一个空格, 程序编制过程中要注意区分。

```

#include<stdio.h>
#define Pas_Triangle_mlength 20
int main(void)
{
    int num[Pas_Triangle_mlength][Pas_Triangle_mlength];
    int i, j, k;
    int n;

    for (i = 0; i < Pas_Triangle_mlength; i++)
    {
        num[i][0] = 1;
        num[i][i] = 1;
        for (j = 1; j < i; j++)
            num[i][j] = num[i][j - 1] * (i - j + 1) / j;
    } //初始化Ci,j。
    scanf("%d", &n);
    getchar();
    while (n <= 12 && n >= 1)
    {
        for (i = 0; i < n; i++)
        {
            for (k = 2 * (n - i - 1); k > 0; k--)
                putchar(' ');

```

```

        for (j = 0; j <= i; j++)
            printf("%-4d", num[i][j]);
        putchar('\n');
    }

    putchar('\n');
    scanf("%d", &n);
    getchar();
}
}

```

```

C:\Users\10334\Documents\Untitled1.exe
    1
  1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
Process returned 10 (0xA)   execution time : 0.286 s
Press any key to continue.

```

(2)编写一个程序,将用户输入的任意正整数逆转,例如,输入 1234,输出 4321。

/*正整数逆转*/

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

```
int Digit(int x);
```

```
int main(void)
```

```
{
```

```
    int x, i, j;
```

```
    int num[20];
```

```
    int newNumber;
```

```
    int lenth_of_x;
```

```
    scanf("%d", &x);
```

```
    while (x)
```

```
    {
```

```
        lenth_of_x = Digit(x);
```

```
        for (i = 0; i <= lenth_of_x; i++)
```

```
        {
```

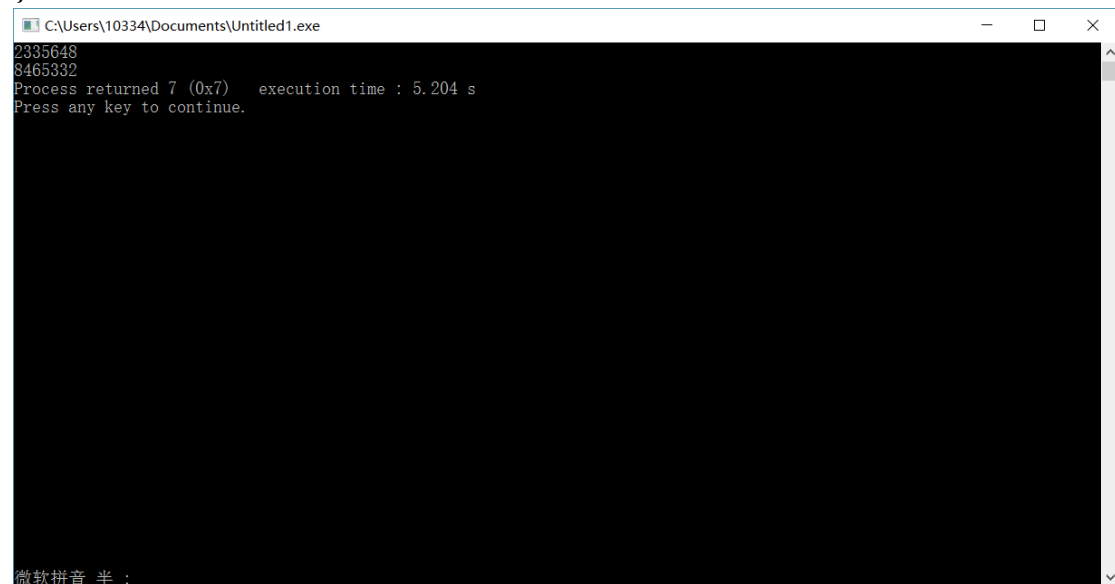
```
            num[i] = x % 10; //i=0,1,2,3...分别取x的个位,十位,百位,千位数....
```

```
            x = x / 10;
```

```

        //printf("%d\n", num[i]);
    }
    for (i = lenth_of_x, j = 1, newNumber = 0; i >= 0; i--, j *= 10)
    {
        num[i] *= j;
        newNumber += num[i];
    }
    printf("%d\n", newNumber);
    scanf("%d", &x);
    getchar();
}
}
int Digit(int x)
{
    int i, count;
    count = 0;
    i = 10;
    while (x / i)
    {
        count++;
        i *= 10;
    } //判断输入数字位数
    return count;
}

```



```

C:\Users\10334\Documents\Untitled1.exe
2335648
8465332
Process returned 7 (0x7)   execution time : 5.204 s
Press any key to continue.

```

(3). 假设工资税金按以下方法计算 $X \leq 1000$ 元, 不收取税金。 $1000 \leq x < 2000$, 收取 5% 的税金; $2000 \leq x < 3000$ 。收取 10% 的税金; $3000 \leq x < 4000$ 。收取 15% 的税金; $4000 \leq x < 5000$ 。收取 20% 的税金; $x > 5000$ 收取 25% 的税金. 编写一个程序, 输入工资额输出应收取税金额度。要求分别用 if 语句和 switch 语句来实现。

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

```

#define Tax_level1 0
#define Tax_level2 0.05
#define Tax_level3 0.10
#define Tax_level4 0.15
#define Tax_level5 0.20
#define Tax_level6 0.25
#define Tax_Boundary1 1000
#define Tax_Boundary2 2000
#define Tax_Boundary3 3000
#define Tax_Boundary4 4000
#define Tax_Boundary5 5000
#define TAX1 (Tax_Boundary2-Tax_Boundary1)*Tax_level2//1000-2000的税
#define TAX2 (Tax_Boundary3-Tax_Boundary2)*Tax_level3+TAX1//1000-3000的
税
#define TAX3 (Tax_Boundary4-Tax_Boundary3)*Tax_level4+TAX2//1000-4000的
税
#define TAX4 (Tax_Boundary5-Tax_Boundary4)*Tax_level5+TAX3//1000-5000的
税

```

```

double If_Method(double wage);

```

```

double Switch_Method(double wage);

```

```

int main(void)

```

```

{

```

```

    double wage;

```

```

    scanf("%lf", &wage);

```

```

    while (wage != 0)

```

```

    {

```

```

        printf("if:%lf\n", If_Method(wage));

```

```

        printf("switch:%lf\n", Switch_Method(wage));

```

```

        scanf("%lf", &wage);

```

```

    }

```

```

}

```

```

double If_Method(double wage)

```

```

{

```

```

    double tax;

```

```

    if (wage <= Tax_Boundary1) tax = Tax_level1*wage;

```

```

    else if (wage <= Tax_Boundary2) tax = Tax_level2*(wage - Tax_Boundary1);

```

```

    else if (wage <= Tax_Boundary3) tax = Tax_level3*(wage - Tax_Boundary2) +
TAX1;

```

```

    else if (wage <= Tax_Boundary4) tax = Tax_level4*(wage - Tax_Boundary3) +
TAX2;

```

```

    else if (wage <= Tax_Boundary5) tax = Tax_level5*(wage - Tax_Boundary4) +

```



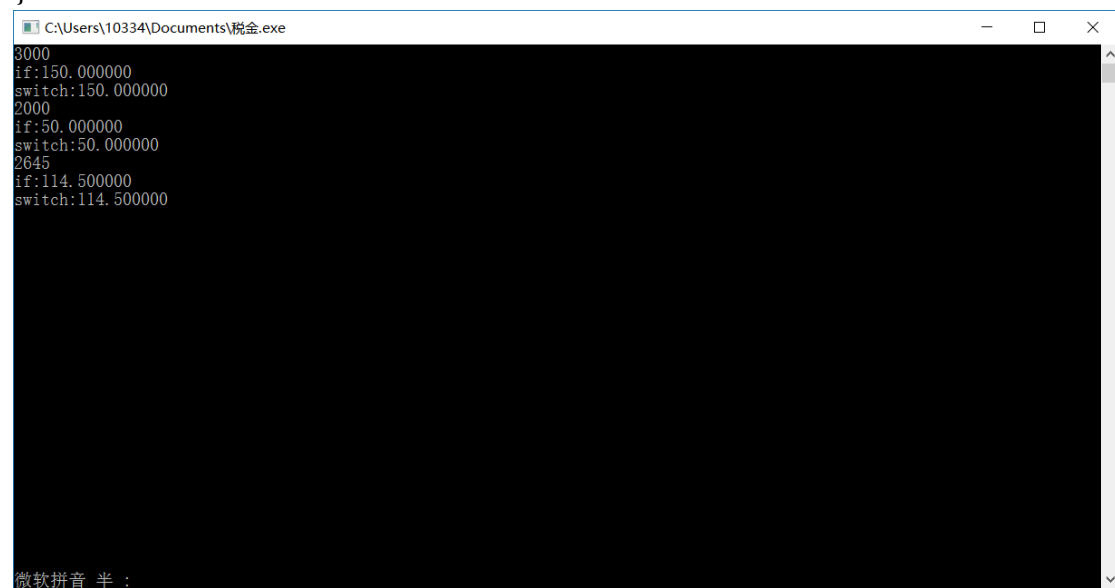
```

TAX3;
    else tax = Tax_level6*(wage - 5000) + TAX4;

    return tax;
}
double Switch_Method(double wage)
{
    double tax;
    switch ((int)(wage - 1) / 1000)
    {
        case 0:tax = Tax_level1*wage; break;
        case 1:tax = Tax_level2*(wage - Tax_Boundary1); break;
        case 2:tax = Tax_level3*(wage - Tax_Boundary2) + TAX1; break;
        case 3:tax = Tax_level4*(wage - Tax_Boundary3) + TAX2; break;
        case 4:tax = Tax_level5*(wage - Tax_Boundary4) + TAX3; break;
        case 5:tax = Tax_level6*(wage - 5000) + TAX4; break;

        default:
            break;
    }
    return tax;
}

```



```

C:\Users\10334\Documents\税金.exe
3000
if:150.000000
switch:150.000000
2000
if:50.000000
switch:50.000000
2645
if:114.500000
switch:114.500000
微软拼音 半:

```

(4) 编写一个程序，用牛顿迭代法求方程 $f(x)=3x^3-4x^2-5x+13=0$ 满足精度 0.000001 的一个近似根，并在屏幕上输出所求近似根。

```

#include<stdio.h>
#include<math.h>
double FUNCTION_F(double x);
double Derivative_F(double x);
#define PRECISION 0.000001
int main(void)

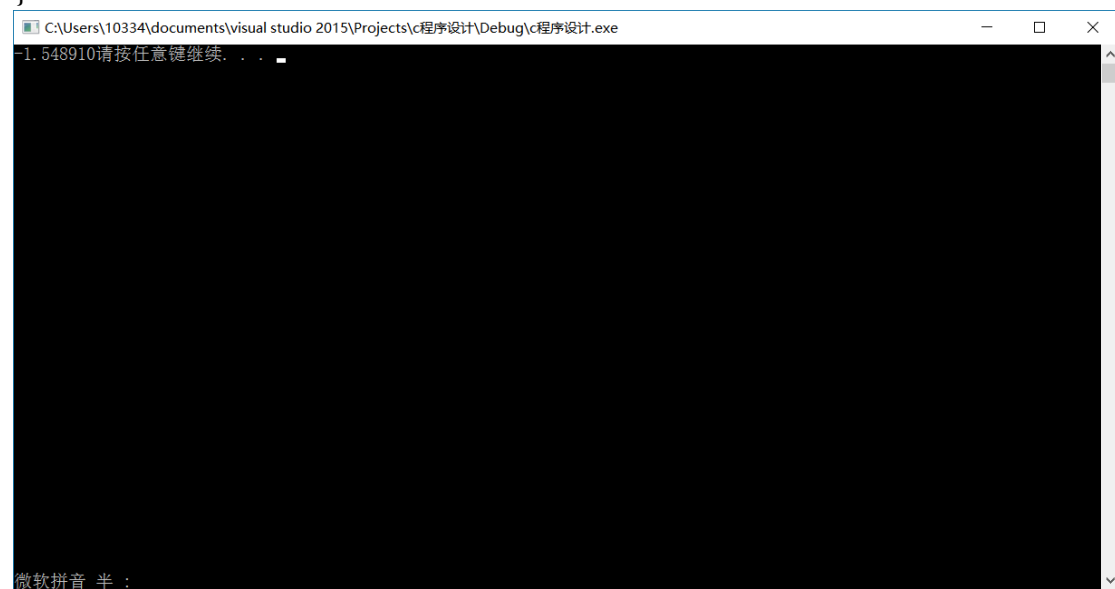
```

```

{
    double x1, x2;
    x1 = 1;
    x2 = 1;
    while (fabs(-FUNCTION_F(x1)/Derivative_F(x1))>PRECISION)
    {
        x1 = x2;
        x2 = x1 - FUNCTION_F(x1) / Derivative_F(x1);

    }
    printf("%lf", x2);
    system("pause");
}
double FUNCTION_F(double x)
{
    return 3 * x*x*x - 4 * x*x - 5 * x + 13;
}
double Derivative_F(double x)
{
    return 9 * x*x - 8 * x - 5;
}

```



(5) 编写一个程序，将输入的一行字符复制到输出，复制过程中，将一个以上的空格字符用一个空格代替。

/*空格处理*/

```

#include<stdio.h>
#include<string.h>
int main(void)
{

```

```

    int N;
    int i,j;

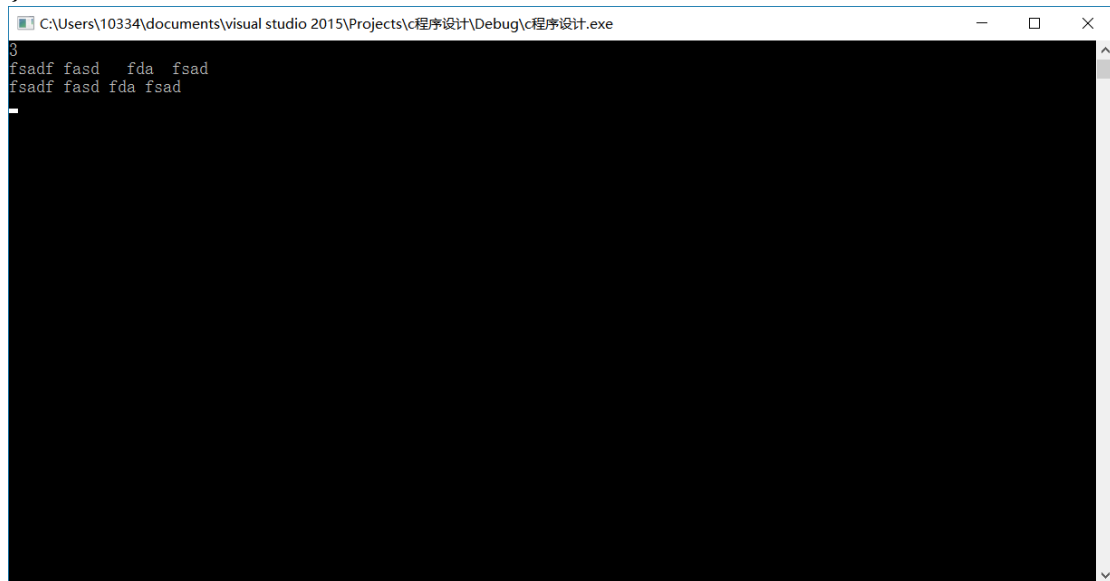
```

```

char mystring[100];

scanf("%d", &N);
getchar();
for (i = 1; i <= N; i++)
{
    fgets(mystring, 100, stdin);
    for (j = 0; j < strlen(mystring); j++)
    {
        if (mystring[j] == ' ')
        {
            putchar(mystring[j]);
            while (mystring[j] == ' ') j++;
        }
        putchar(mystring[j]);
    }
}
}

```



```

C:\Users\10334\documents\visual studio 2015\Projects\c程序设计\Debug\c程序设计.exe
3
fsadf fsad fda fsad
fsadf fsad fda fsad

```

2.3、实验感想

编写 C 语言程序要熟练掌握复合语句, if 语句, switch 语句的使用, 熟练掌握 for、while、do-while 基本的循环控制语句的使用, 使程序更为流畅, 掌握重复循环技术, 了解转移语句与标号语句。 编写条件语句时, 要充分想到所有的情况, 避免遗漏, 是程序结果出现错误; 编写循环语句时, 要充分考虑循环结束时的条件, 一面出错, 使程序运行不了或出现死循环; 在考虑循环结束的条件时, 应尽量做一个最好的选择, 让程序变得简洁, 避免占用过长的篇幅, 同时也可以减少错误的发生; 要了解 for 语句、while 语句、do-while 语句之间以及 if 语句与 switch 语句之间的转换, 使编写程序时有更大的选择空间, 从而找到一个最优的方法。 还要熟练掌握集成开发环境中的调试功能, 例如单步执行, 设置断点等

等。以免因为修改程序中出现的错误而消耗过多的时间。