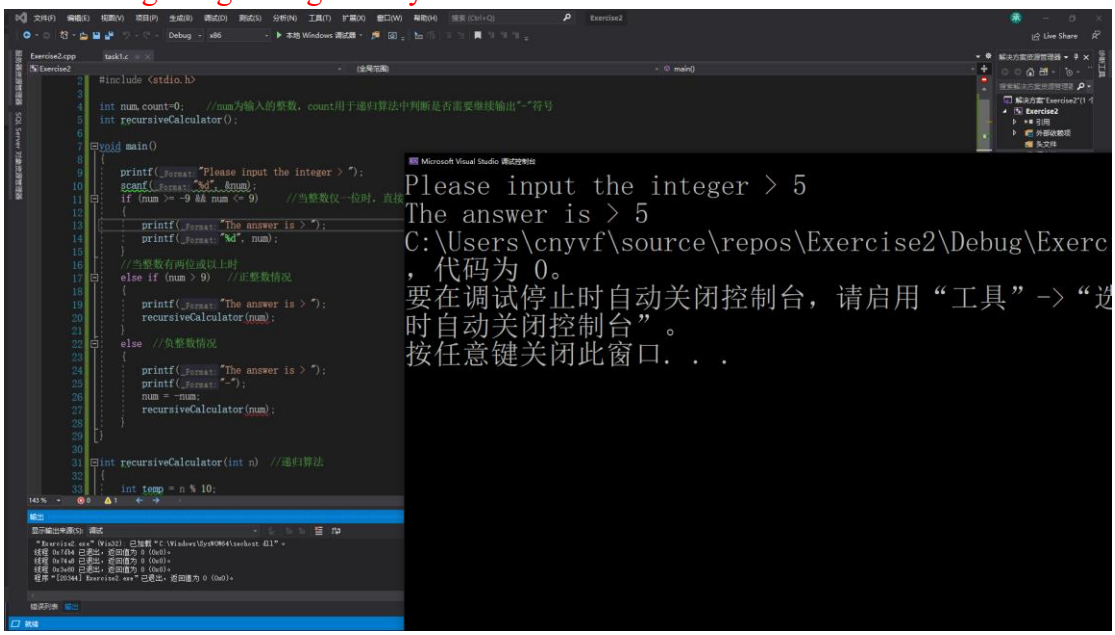
Exercise 1

1. Using a recursive method to transform a 5-bit integer n into a string consists of its digit, for example, if you input 48693, the output information should be 4-8-6-9-3. The integer n must be input from the keyboard, and this number could be any value.

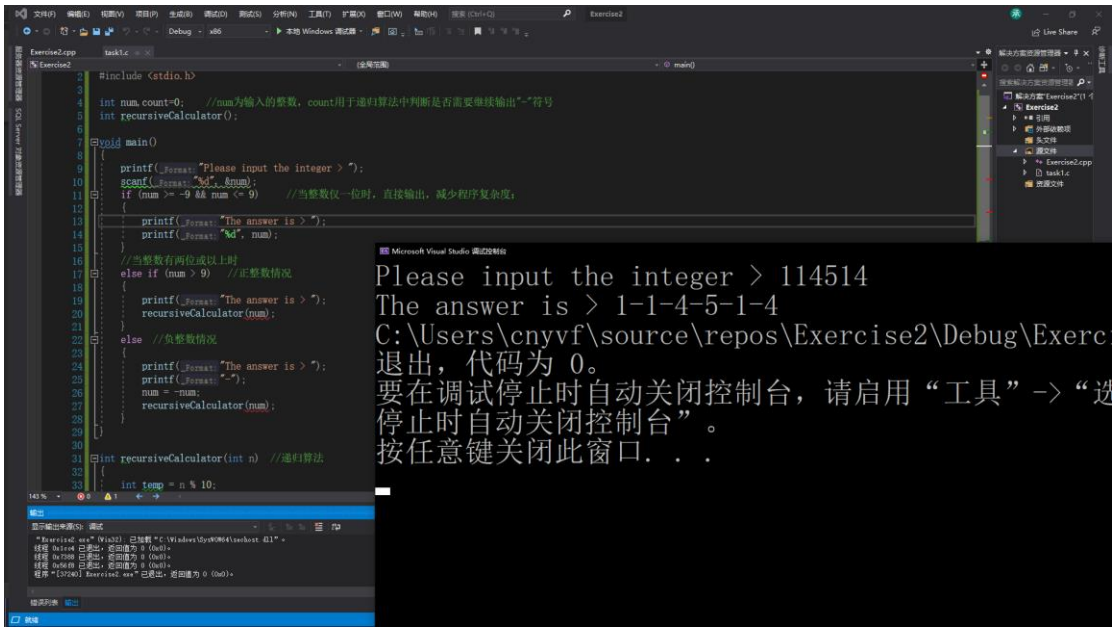
```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int num,count=0;    //num为输入的整数，count用于递归算法中判断是否需要继续输出"-"符号
int recursiveCalculator();

void main()
{
    printf("Please input the integer > ");
    scanf("%d", &num);
    if (num >= -9 && num <= 9)      //当整数仅一位时，直接输出，减少程序复杂度；
    {
        printf("The answer is > ");
        printf("%d", num);
    }
    //当整数有两位或以上时
    else if (num > 9)   //正整数情况
    {
        printf("The answer is > ");
        recursiveCalculator(num);
    }
    else  //负整数情况(我们将负号看作一位处理)
    {
        printf("The answer is > ");
        printf("--");
        num = -num;
        recursiveCalculator(num);
    }
}

int recursiveCalculator(int n)  //递归算法
{
    int temp = n % 10;
    n = n / 10;
    count++;
```

```
    if(n!=0)     //若输入的整数尚未完全化完，则继续进入该函数进行运算（该步得到的值为逆序）
    {
        recursiveCalculator(n);
    }

    //运算完成后逐步返回进行倒置输出，通过count计算递进次数并以此判断是否为末尾，是否需要输出"-"
    count--;
    if(count!=0)
    {
        printf("%d-", temp);
    }
    else
    {
        printf("%d", temp);
    }
    return 0;
}
```

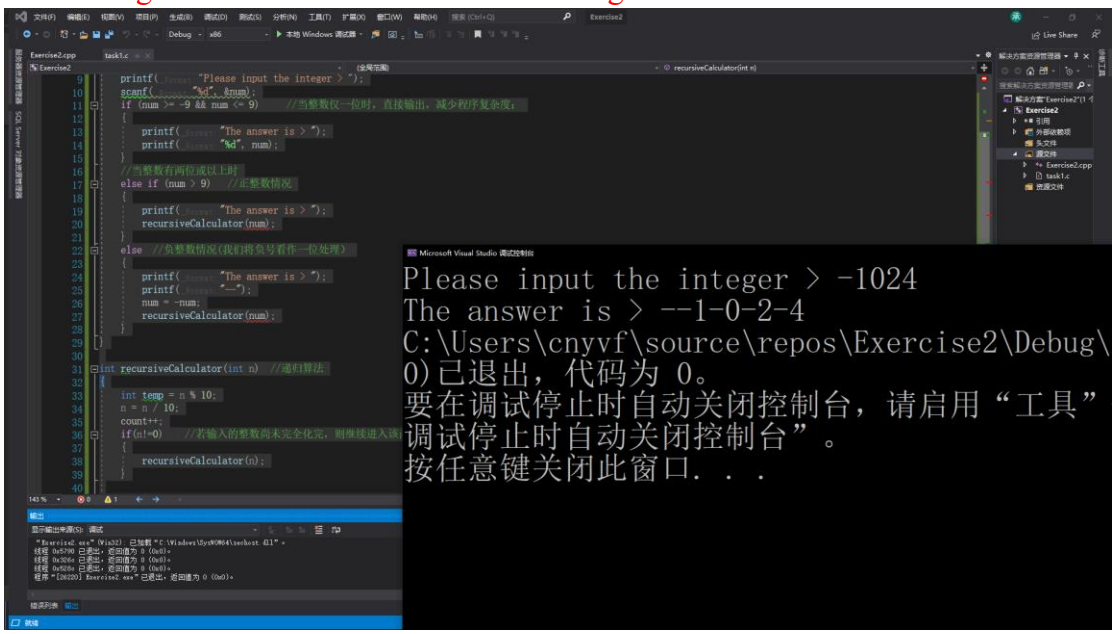
Case1: Single-digit integers only



Case2:Positive numbers with at least two integers



Case3: Negative numbers with at least two integers



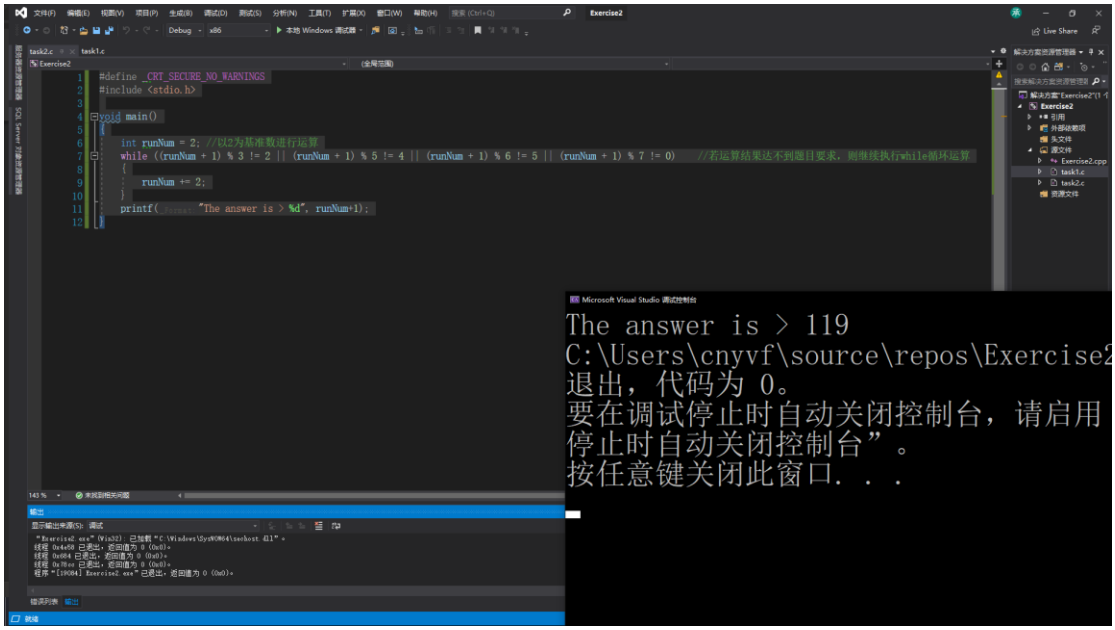
Exercise 2

2. Programming to solve Einstein's math problems. Einstein had a math problem like this, a long ladder, if each step on the 2 order, the last remaining 1 orders, if each step on 3 orders, the last remaining 2 orders, if each step on the 5 order, the last remaining 4 orders, if each step on the 6 order, the last remaining 5 orders, only each step on the 7 order, the last just one order is not left, please ask the ladder

(编程解决爱因斯坦数学题。爱因斯坦曾出过这样一道数学题，一条长阶梯，若每步上2阶，最后剩下1阶，若每步上3阶，最后剩下2阶，若每步上5阶，最后剩下4阶，若每步上6阶，最后剩下5阶，只有每步上7阶，最后刚好一阶也不剩下，请问该阶梯至少有多少阶)

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

void main()
{
    int runNum = 2;   //以2为基准数进行运算
    while ((runNum + 1) % 3 != 2 || (runNum + 1) % 5 != 4 || (runNum + 1) % 6 != 5 || (runNum + 1) % 7 != 0)//若运算结果达不到题目要求，则继续执行while循环运算
    {
        runNum += 2;
    }
    printf("The answer is > %d", runNum+1);
}
```

Case1:



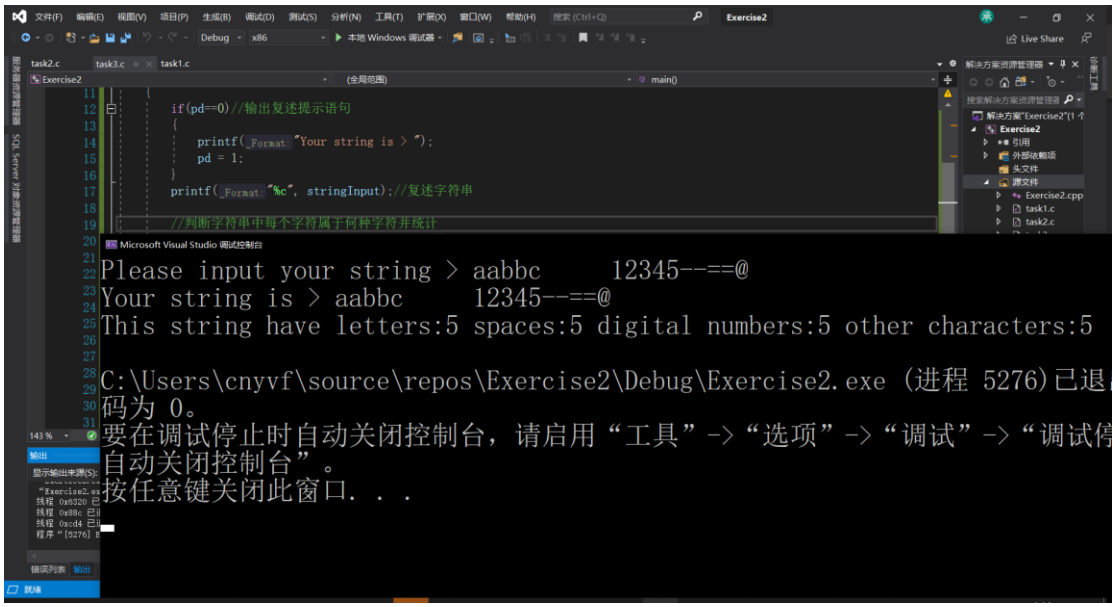
Exercise 3

3. Programming uses the while loop statement, and use the function getchar () to receive a string of characters, and counts the number of letters, the number of spaces, the number of digital numbers, and the number of other characters. Output the results.

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

void main()
{
    char stringInput;
    int numLetters = 0, numSpaces = 0, numDigital = 0, numOther = 0; //分别表示字母数，空格数，数字数，其他字符数
    int pd = 0; //用于判断是否输出了复述字符串提示语句
    printf("Please input your string > ");
    while((stringInput = getchar())!='\n') //使用getchar获取字符串直至按下回车键，循环每一个字符
    {
        if(pd==0)//输出复述提示语句
        {
            printf("Your string is > ");
            pd = 1;
        }
        printf("%c", stringInput);//复述字符串

        //判断字符串中每个字符属于何种字符并统计
        if((stringInput>='A'&&stringInput<='Z')|| (stringInput >= 'a' && stringInput <= 'z'))
        {
            numLetters++;
        }
        else if (stringInput==' ')
        {
            numSpaces++;
        }
        else if (stringInput >= '0' && stringInput <= '9')
        {
```

```
            numDigital++;
        }
        else
        {
            numOther++;
        }
    }
    //输出结果
    printf("\nThis string have letters:%d spaces:%d digital numbers:%d other characters:%d", numLetters, numSpaces, numDigital, numOther);
    printf("\n");
}
```

Case1:



Exercise 4

4. Calculate the result of the following equation by using the **while statements**:

$$Sn = a + aa + aaa + aaaa + \underbrace{a.....a}_{n\uparrow a}$$

where a and n are any given integer numbers, and you must input them from the keyboard.

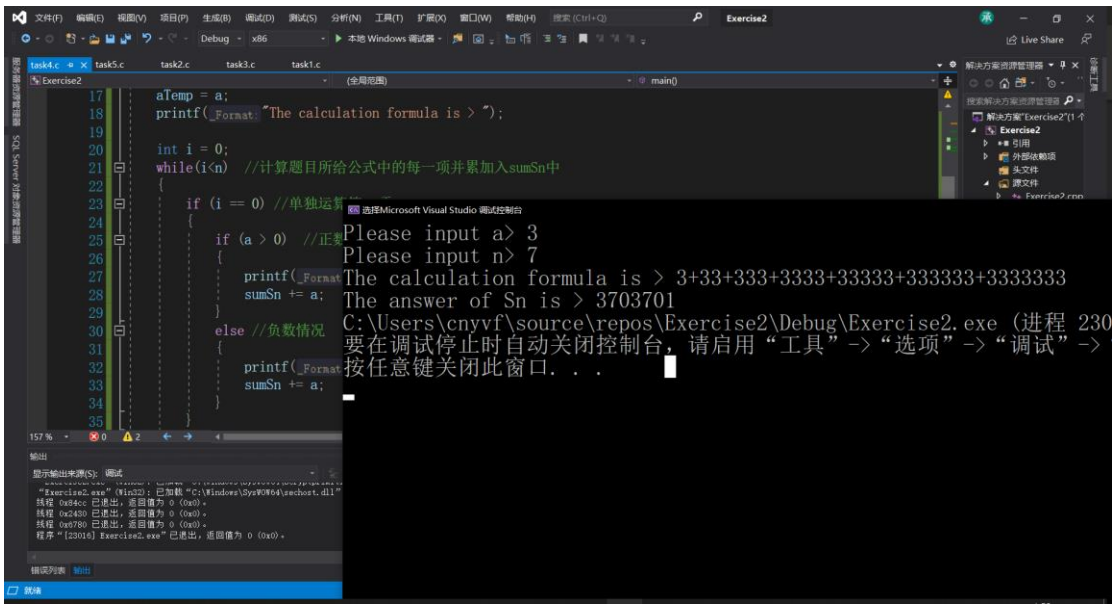
For example, if you input a=3, and n=7, then you need to calculate:

$$3 + 33 + 333 + 3333 + 33333 + 333333 + 3333333$$

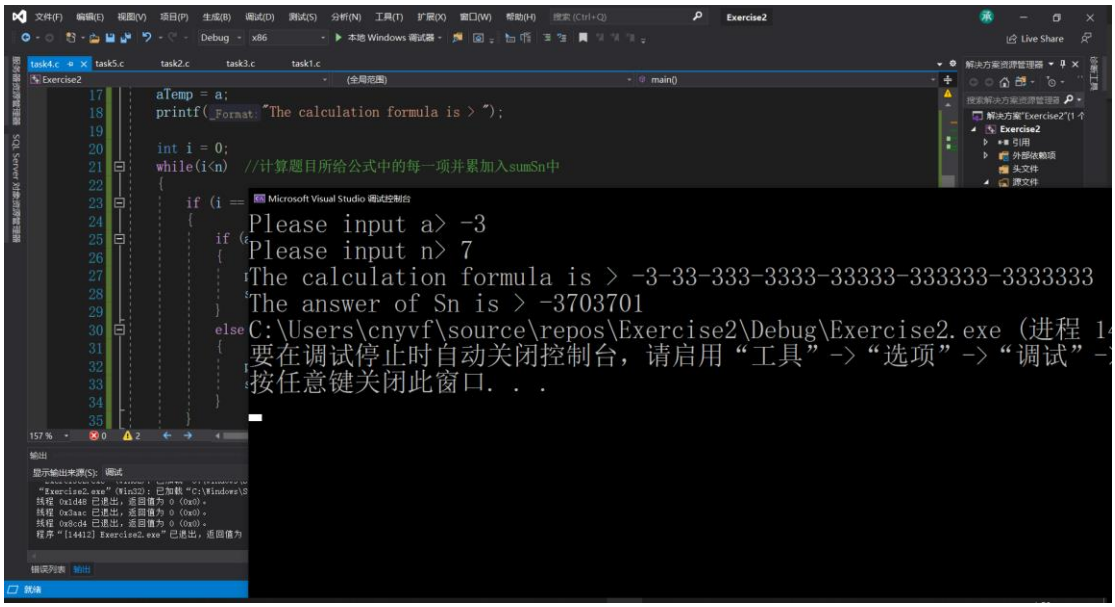
```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

void main()
{
    int sumSn = 0, a, n, count = 0, aTemp;  //总值，初始值，项的个数，整数位数，临时计数器
    printf("Please input a> ");
    scanf("%d", &a);
    printf("Please input n> ");
    scanf("%d", &n);
    aTemp = a;
    while (aTemp != 0)//求整数有几位
    {
        aTemp /= 10;
        count++;
    }
    aTemp = a;
    printf("The calculation formula is > ");

    int i = 0;
    while(i<n)    //计算题目所给公式中的每一项并累加入sumSn中 (while)
    {
        if (i == 0) //单独运算第一项
        {
            if (a > 0)    //正数情况
            {
```

```
                printf("%d+", a);
                sumSn += a;
            }
            else //负数情况
            {
                printf("%d", a);
                sumSn += a;
            }
        }
        else
        {
            a = a * 10 + aTemp; //用于计算下一项
            sumSn += a; //累加
            if (a >= 0) //正数情况
            {
                if (i != n - 1)
                {
                    printf("%d+", a);
                }
                else
                {
                    printf("%d", a);
                }
            }
            else //负数情况
            {
                printf("%d", a);
            }
        }
        i++;
    }
    printf("\nThe answer of Sn is > %d", sumSn);
}
```

Case1: a is a positive number



Case2: a is a negative number



Exercise 5

5. Calculate the result of the following equation by using the **for statements**:

$$Sn = a + aa + aaa + aaaa + \underbrace{a.....a}_{n个a}$$

where a and n are any given integer numbers, and you must input them from the keyboard.

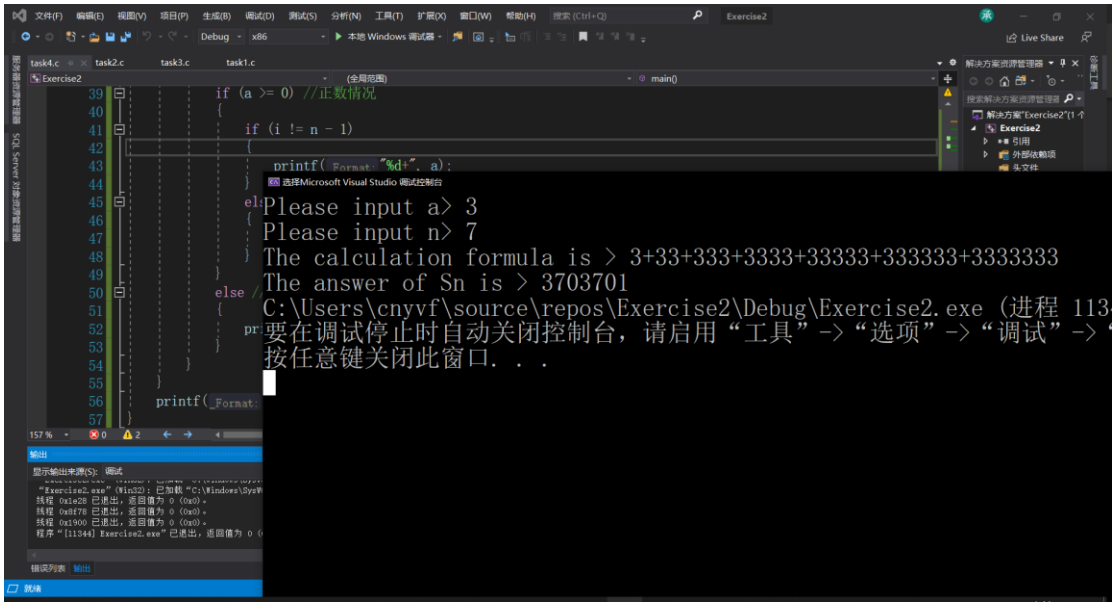
For example, if you input a=3, and n=7, then you need to calculate:

3 + 33 + 333 + 3333 + 33333 + 333333 + 3333333

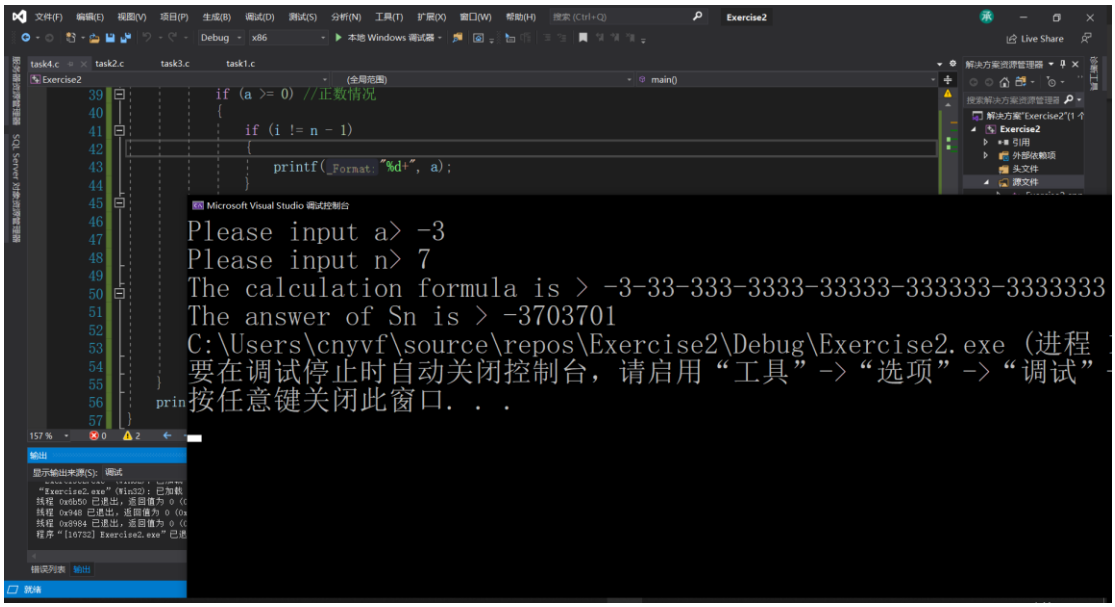
```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

void main()
{
    int sumSn=0, a, n,count=0,aTemp;    //总值，初始值，项的个数，整数位数，临时计数器
    printf("Please input a> ");
    scanf("%d", &a);
    printf("Please input n> ");
    scanf("%d", &n);
    aTemp = a;
    while(aTemp!=0)//求整数有几位
    {
        aTemp /= 10;
        count++;
    }
    aTemp = a;
    printf("The calculation formula is > ");

    for(int i=0;i<n;i++)  //计算题目所给公式中的每一项并累加入sumSn中（for）
    {
        if(i==0) //单独运算第一项
        {
            if(a>0)  //正数情况
            {
                printf("%d+", a);
                sumSn += a;
            }
            else //负数情况
```

```
            {
                printf("%d", a);
                sumSn += a;
            }
        }
        else
        {
            a = a * 10 + aTemp; //用于计算下一项
            sumSn += a; //累加
            if (a >= 0) //正数情况
            {
                if (i != n - 1)
                {
                    printf("%d+", a);
                }
                else
                {
                    printf("%d", a);
                }
            }
            else //负数情况
            {
                printf("%d", a);
            }
        }
    }
    printf("\nThe answer of Sn is > %d", sumSn);
}
```

Case1: a is a positive number



Case2: a is a negative number



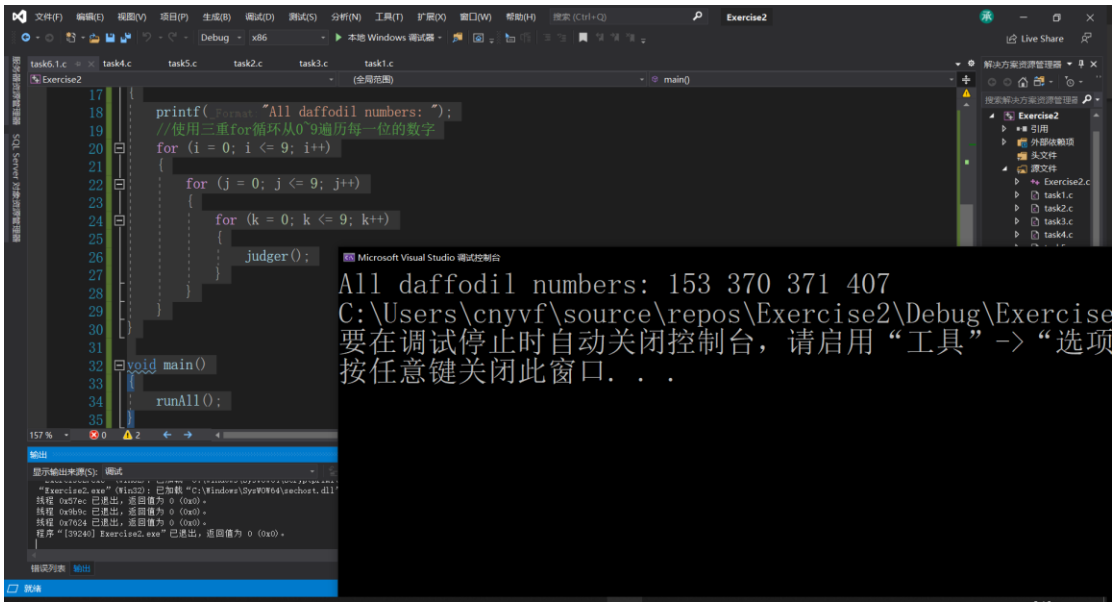
Exercise 6 method_1

6. Find all daffodil numbers by at least two methods with **for statements**.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <math.h>

int i, j, k;

void judger()
{
    //判断每个位的数字相加是否等于其本身，判断该数字是否为三位数
    if (pow(i, 3) + pow(j, 3) + pow(k, 3) == i * 100 + j * 10 + k && pow(i, 3) + pow(j, 3) + pow(k, 3) >= 100 && pow(i, 3) + pow(j, 3) + pow(k, 3) <= 999)
    {
        printf("%d%d%d ", i, j, k);    //输出水仙花数字
    }
}

void runAll()
{
```

```
    printf("All daffodil numbers: ");
    //使用三重for循环从0~9遍历每一位的数字
    for (i = 0; i <= 9; i++)
    {
        for (j = 0; j <= 9; j++)
        {
            for (k = 0; k <= 9; k++)
            {
                judger();
            }
        }
    }
}

void main()
{
    runAll();
}
```

Case1:



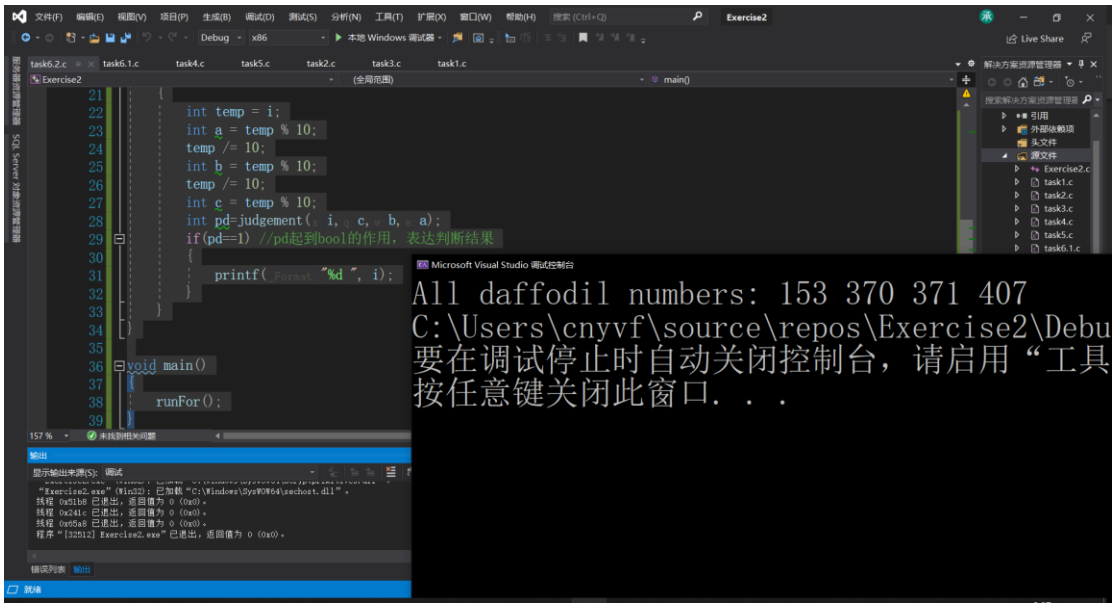
Exercise 6 method_2

6. Find all daffodil numbers by at least two methods with **for statements**.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <math.h>

int judgement(int x,int q,int w,int e) //判断函数，用于判断每个位的数字相加是否等于其本身
{
    if (pow(q, 3) + pow(w, 3) + pow(e, 3) == x)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

void runFor()
{
    printf("All daffodil numbers: ");
    for(int i=100;i<=999;i++) //遍历每一个三位数
    {
        int temp = i;
        int a = temp % 10;
        temp /= 10;
        int b = temp % 10;
        temp /= 10;
        int c = temp % 10;
        int pd=judgement(i,c,b,a);
        if(pd==1) //pd起到bool的作用，表达判断结果
        {
            printf("%d ", i);
        }
    }
}

void main()
{
```

```
    runFor();
}
```

Case1:



Exercise 7

7. Write a program to find all the perfect number under 1000, where the perfect number is a number such that the summation of all its factors is equal to its own value, and then output the result in the following format:

   6: its factors are 1, 2, 3

   x: its factors are .....

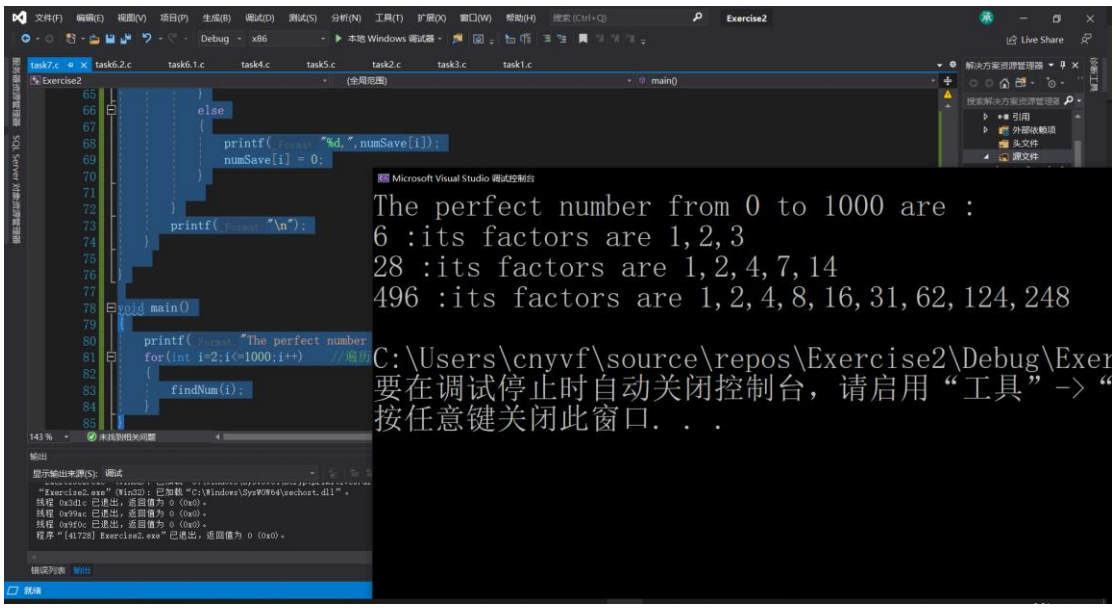
   .....

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include<stdlib.h>

int comp(const void* a, const void* b)//比较函数，用于给数组排序
{
    return *(int*)a - *(int*)b;
}

void findNum(int x)
{
    int temp = 0;
```

```
int numSave[1000];
int count = 0;
//寻找x的每一个真因子并累加
for(int i=1;i<=x;i++)
{
    for(int j=i;j<=x;j++)
    {
        if(i*j==x)
        {
            if(i==j) //避免重复操作
            {
                temp += i;
                numSave[count] = i;
                count++;
            }
            else
            {
                temp = temp + i + j;
                if(i!=x&&j!=x)
                {
                    numSave[count] = i;
                    count++;
                    numSave[count] = j;
                    count++;
                }
                else if(i==x)//避免加入其本身
                {
                    numSave[count] = j;
                    count++;
                }
                else
                {
                    numSave[count] = i;
                    count++;
                }

            }
        }
    }
}
temp = temp - x;

if(temp==x)  //判断x的真因子之和是否与其相等
{
```

```
        qsort(numSave, count, sizeof(int), comp); //qsort函数，用于给数组排序
        printf("%d :its factors are ", x);
        for(int i=0;i<count;i++)
        {
            if(i==count-1)
            {
                printf("%d",numSave[i]);
                numSave[i] = 0;
            }
            else
            {
                printf("%d,",numSave[i]);
                numSave[i] = 0;
            }

        }
        printf("\n");
    }

}

void main()
{
    printf("The perfect number from 0 to 1000 are : \n");
    for(int i=2;i<=1000;i++)  //遍历一千以内的每一个可能是完全数的数
    {
        findNum(i);
    }
}
```

Case1:



Exercise 8

8. Enter an integer, calculate the summation of the squares of the inverse position digits of this integer, as the following example.

从键盘输入一个整数，求该数各位上的数字分别相逆位的平方和，以及总和。比如：

Input a number: 6825739

9*9 + 6*6 = 117, Sum = 117

3*3 + 8*8 = 73, Sum = 190

7*7 + 2*2 = 53, Sum = 243

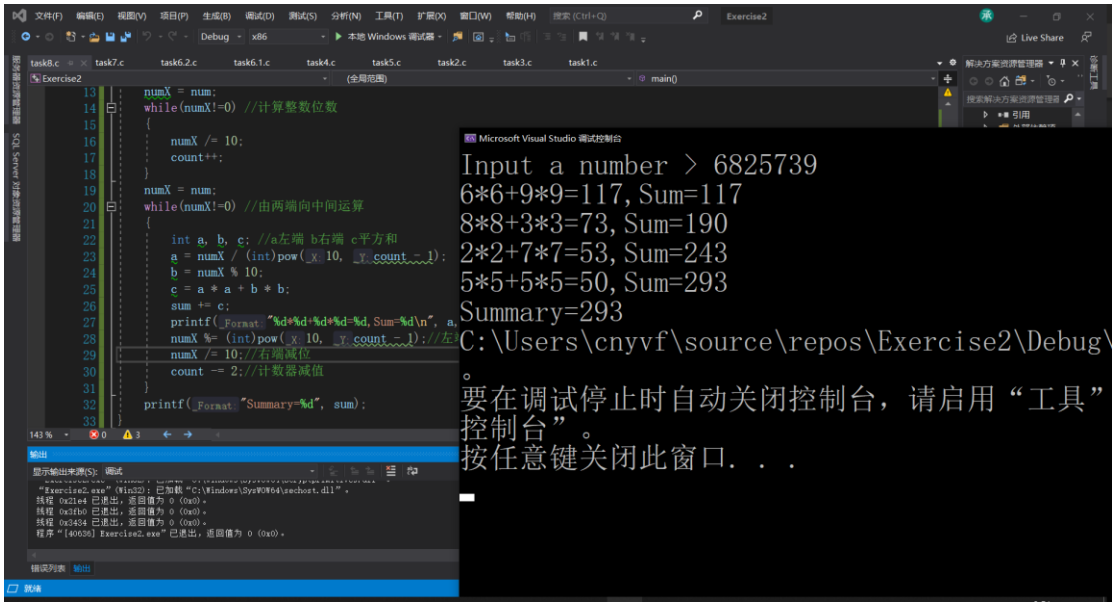
5*5 + 5*5 = 50, Sum = 293

Summary = 293

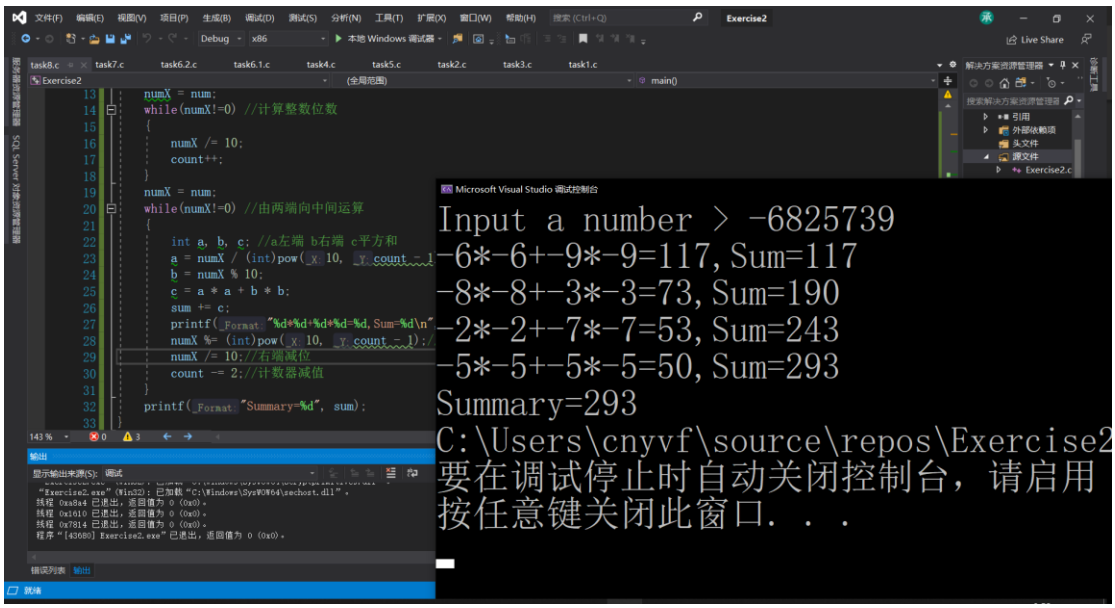
```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <math.h>

void main()
{
    int num;
    int numX;
    int count = 0;
    int sum = 0;
    printf("Input a number > ");
```

```
    scanf("%d", &num); //输入整数
    numX = num;
    while(numX!=0) //计算整数位数
    {
        numX /= 10;
        count++;
    }
    numX = num;
    while(numX!=0) //由两端向中间运算
    {
        int a, b, c; //a左端 b右端 c平方和
        a = numX / (int)pow(10, count - 1);
        b = numX % 10;
        c = a * a + b * b;
        sum += c;
        printf("%d*%d+%d*%d=%d,Sum=%d\n", a, a, b, b, c, sum);
        numX %= (int)pow(10, count - 1);//左端减位
        numX /= 10;//右端减位
        count -= 2;//计数器减值
    }
    printf("Summary=%d", sum);
}
```

Case1: Positive integers



Case2: Negative integer


numX = num;
while(numX!=0) //计算整数位数
{
numX /= 10;
count++;
}
numX = num;
while(numX!=0) //由两端向中间运算
{
int a, b, c; //a左端 b右端 c平方和
a = numX / (int)pow(x: 10, y: count - 1
b = numX % 10;
c = a * a + b * b;
sum += c;
printf(Format: "%d*%d+%d*%d=%d, Sum=%d\n"
numX %= (int)pow(x: 10, y: count - 1);
numX /= 10;//右端减位
count -= 2;//计数器减值
}
printf(Format: "Summary=%d", sum);
Microsoft Visual Studio 调试控制台
Input a number > -6825739
-6*-6+-9*-9=117, Sum=117
-8*-8+-3*-3=73, Sum=190
-2*-2+-7*-7=53, Sum=243
-5*-5+-5*-5=50, Sum=293
Summary=293
C:\Users\cnyvf\source\repos\Exercise2
要在调试停止时自动关闭控制台，请启用
按任意键关闭此窗口. . .