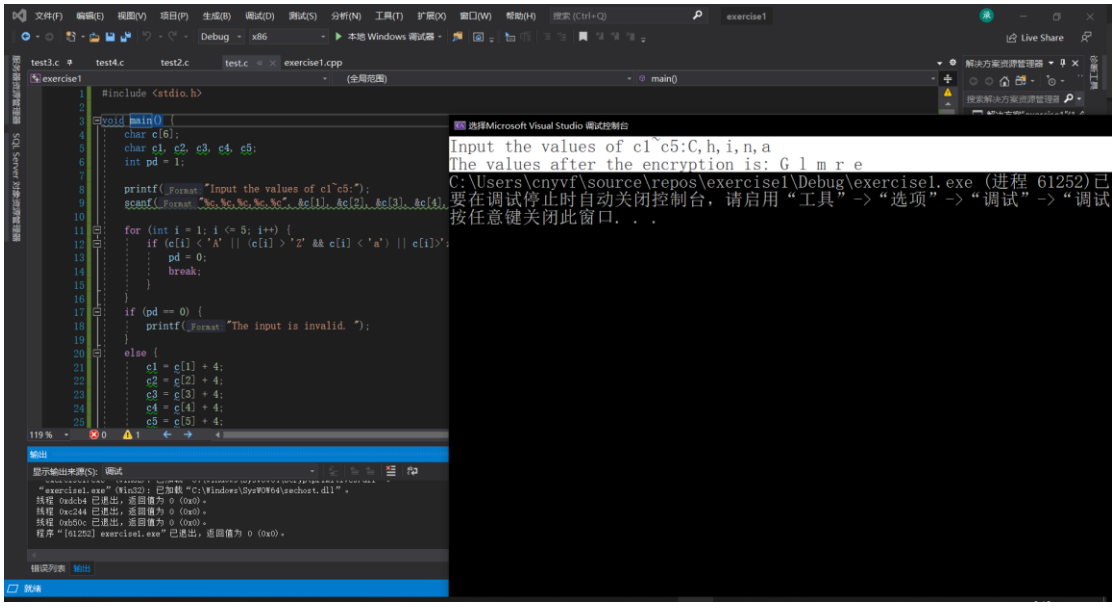
Exercise 1

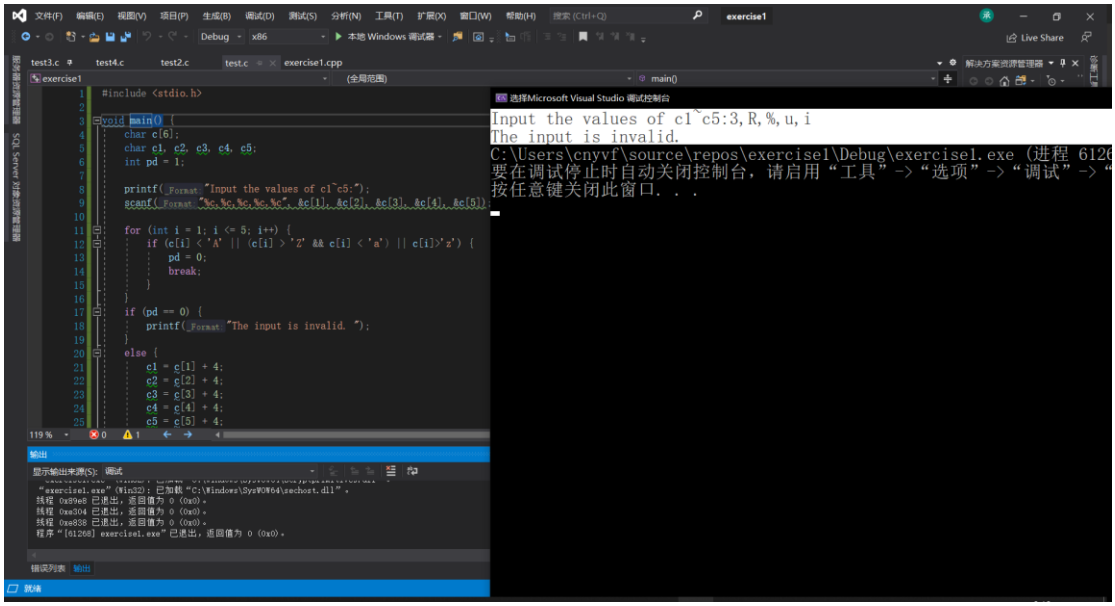
```
#include <stdio.h>

void main()
{
	char c[6];                          // Define an array to store the five characters entered
	char c1, c2, c3, c4, c5;            // Five variables for storing results

	// A bool value used to determine if the input characters are all English letters (1 is true 0 is false)
	int pd = 1;

	printf("Input the values of c1~c5:");
	// Input characters
	scanf("%c,%c,%c,%c,%c", &c[1], &c[2], &c[3], &c[4], &c[5]);

	// The for loop iterates through each character of the input
	for (int i = 1; i <= 5; i++)
	{
		// Determine if the character entered is upper or lower case, if not then pd=0
		if (c[i] < 'A' || (c[i] > 'Z' && c[i] < 'a') || c[i]>'z')
		{
			pd = 0;
			break;
		}
	}
	if (pd == 0)
	{
		// If this variable is 0, then the characters entered are not all English letters
		printf("The input is invalid. ");          // Prompt for input error in this case
	}
	else
	{
		// If the input values are all English letters, the ASCII value of each element of the array is added by 4 and assigned to the corresponding variable
		c1 = c[1] + 4;
		c2 = c[2] + 4;
		c3 = c[3] + 4;
		c4 = c[4] + 4;
		c5 = c[5] + 4;
		// Formatting the output corresponding variables
```

```
            printf("The values after the encryption is: %c %c %c %c %c", c1, c2, c3, c4, c5);
        }
}
```

Case1: The input value is five English letters.



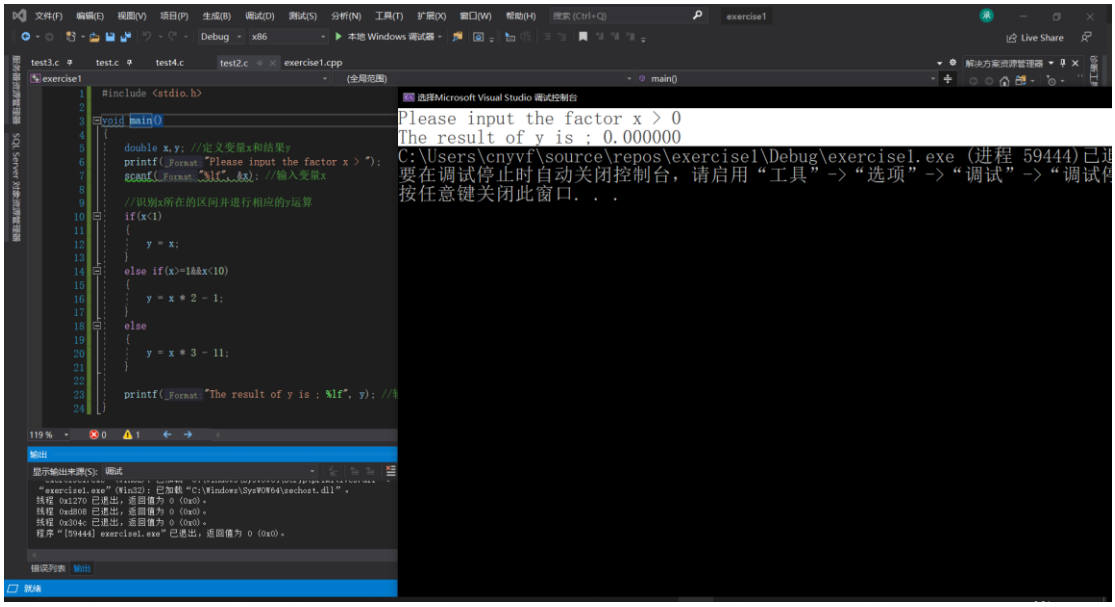
Case2： The input value is not five English letters.



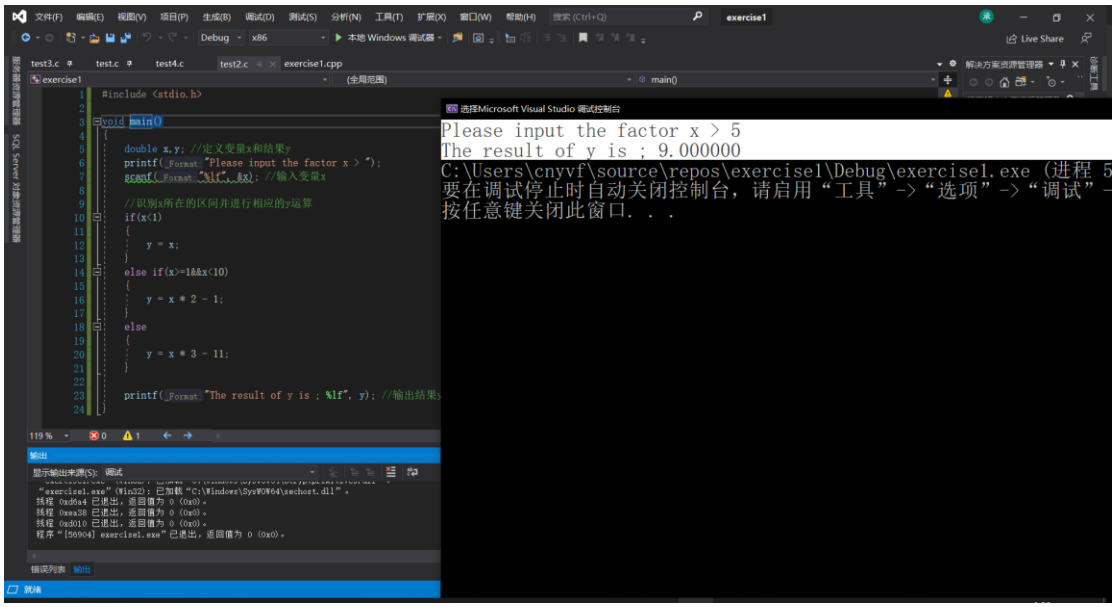
Exercise 2

```c
#include <stdio.h>

void main()
{
    double x,y; // Define the variable x and the result y
    printf("Please input the factor x > ");
    scanf("%lf", &x); // Input variable x

    // Identify the interval where x is located and perform the corresponding y operation
    if(x<1)
    {
        y = x;
    }
    else if(x>=1&&x<10)
    {
        y = x * 2 - 1;
    }
    else
    {
        y = x * 3 - 11;
    }

    printf("The result of y is ; %lf", y); // Output result y
}
```

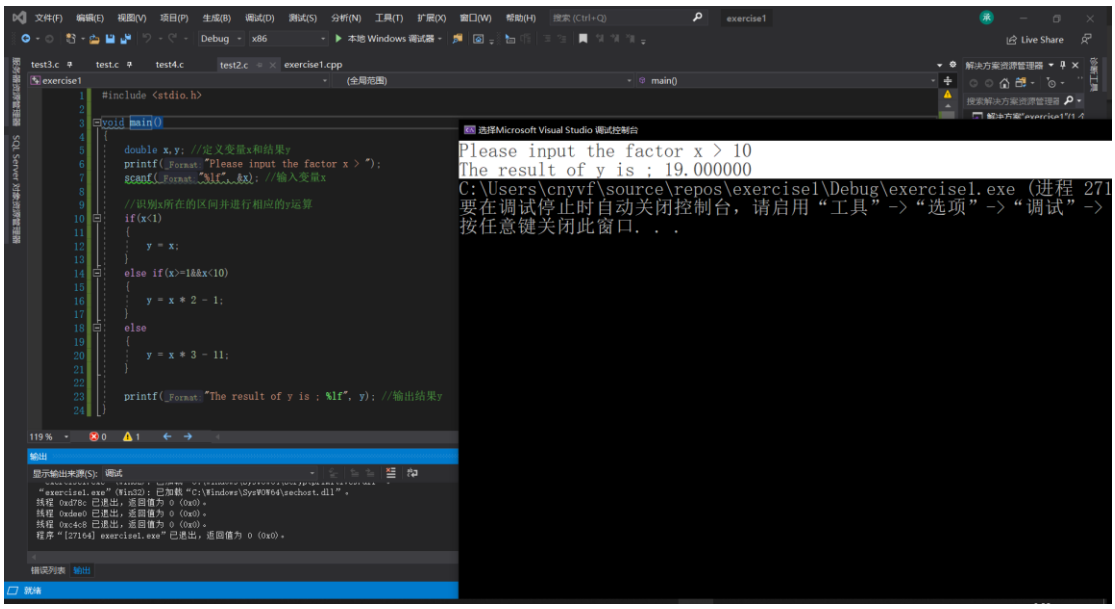
Case1: When x is less than 1



Case2: When x is greater than or equal to one and less than ten



Case3: When x is greater than or equal to 10

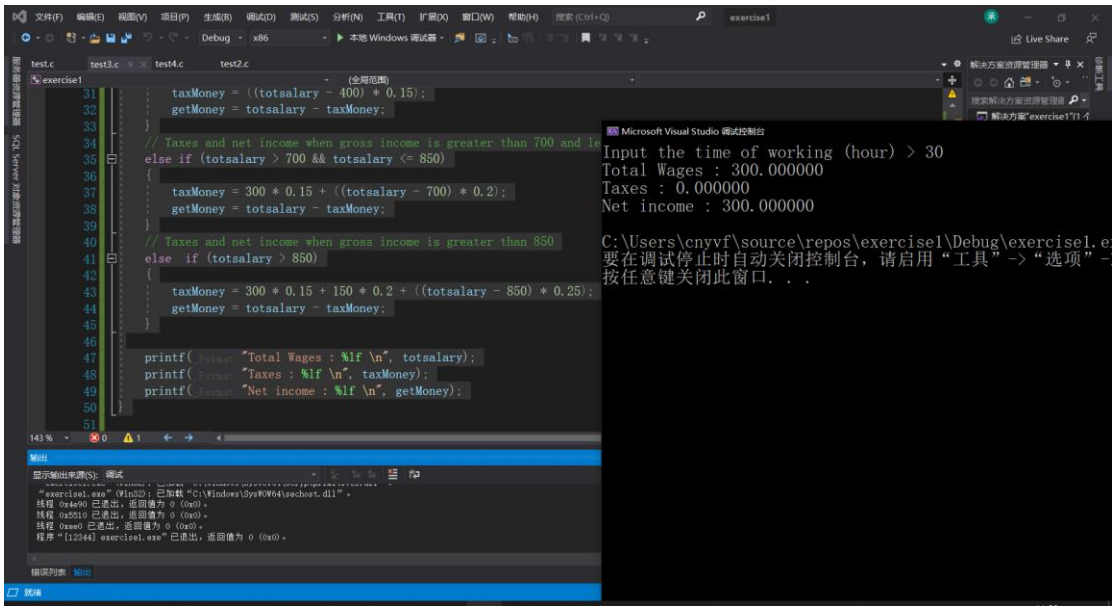


Exercise 3

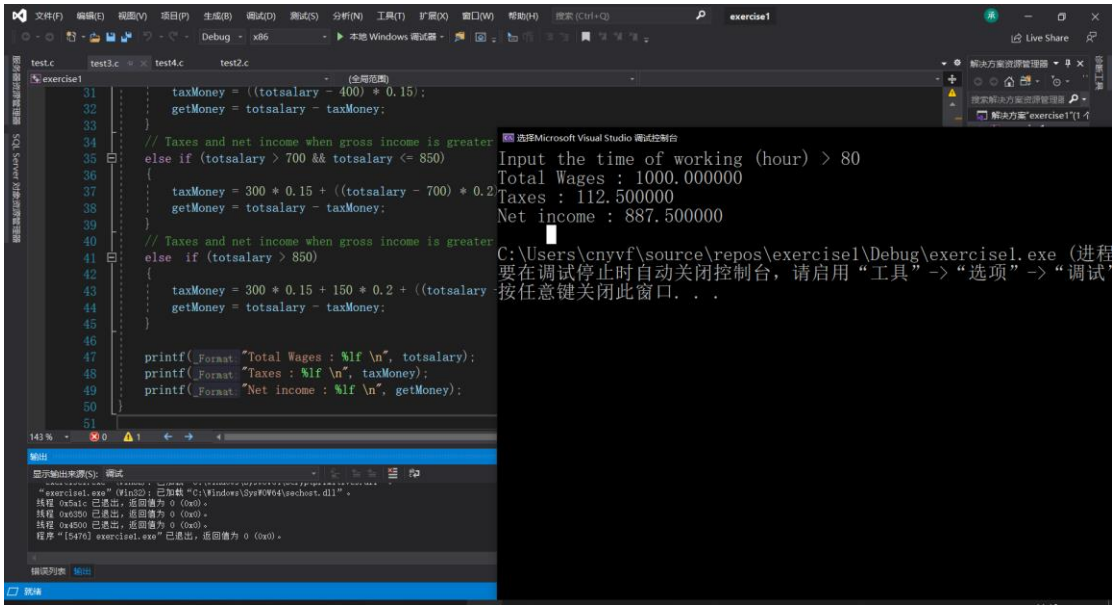
```c
#include <stdio.h>

void main()
{
    double totsalary = 0; // Total salary
    double workTime = 0; // Working hours
    double taxMoney = 0; // Taxes
    double getMoney = 0; // Net income

    printf("Input the time of working (hour) > ");
    scanf("%lf", &workTime); // Input working time
    // Calculate the total revenue when the working hours are greater than
40
    if (workTime > 40)
    {
        totsalary = (40 * 10.0) + ((workTime - 40) * 1.5 * 10.0);
    }
    // Calculate the total revenue when the working hours are less than 40
    else
    {
        totsalary = workTime * 10;
    }
    //Taxes and net income when gross income is less than 400
    if(totsalary<=400)
    {
        taxMoney = 0;
        getMoney = totsalary - taxMoney;
    }
    //Taxes and net income when income is greater than 400 and less than or
equal to 700
    else if (totsalary > 400 && totsalary <= 700)
    {
        taxMoney = ((totsalary - 400) * 0.15);
        getMoney = totsalary - taxMoney;
    }
    // Taxes and net income when gross income is greater than 700 and less
than or equal to 850
    else if (totsalary > 700 && totsalary <= 850)
    {
        taxMoney = 300 * 0.15 + ((totsalary - 700) * 0.2);
```

```
        getMoney = totsalary - taxMoney;
    }
    // Taxes and net income when gross income is greater than 850
    else  if (totsalary > 850)
    {
        taxMoney = 300 * 0.15 + 150 * 0.2 + ((totsalary - 850) * 0.25);
        getMoney = totsalary - taxMoney;
    }
    printf("Total Wages : %lf \n", totsalary);
    printf("Taxes : %lf \n", taxMoney);
    printf("Net income : %lf \n", getMoney);
}
```

Case1: Working hours less than forty



Case2: Working hours more than forty



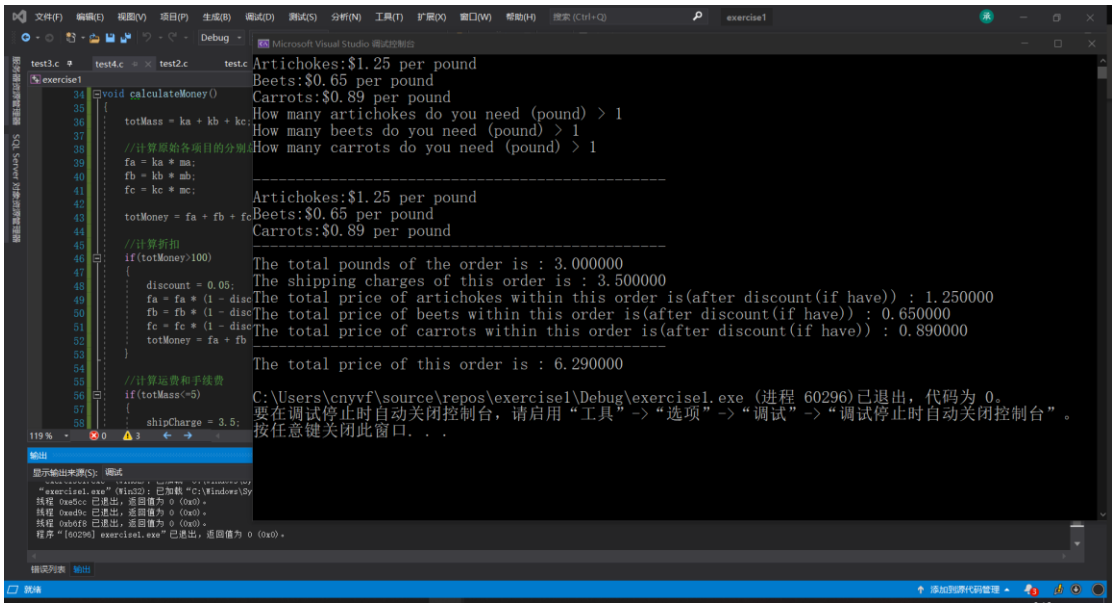
Exercise 4

```
#include <stdio.h>

double ka, kb, kc; // The mass of each type of goods required
double ma, mb, mc; // Unit price of each type of goods
double discount=0; // Discount
double totMoney=0; // Total amount of money needed in the end
double fa, fb, fc; // Final total discounted price per item
double totMass=0;    // Total mass
double shipCharge=0;// Shipping Fee

// Define and display unit price
void printfPerMoney()
{
	ma = 1.25;
	mb = 0.65;
	mc = 0.89;
	printf("Artichokes:$1.25 per pound\n");
	printf("Beets:$0.65 per pound\n");
	printf("Carrots:$0.89 per pound\n");
}

// The user enters the desired amount
void inputMass()
{
	printf("How many artichokes do you need (pound) > ");
	scanf("%lf", &ka);
	printf("How many beets do you need (pound) > ");
	scanf("%lf", &kb);
	printf("How many carrots do you need (pound) > ");
	scanf("%lf", &kc);
}

// Operation process
void calculateMoney()
{
	totMass = ka + kb + kc; // Total mass

	// Calculate the total price of each of the original items
	fa = ka * ma;
	fb = kb * mb;
```

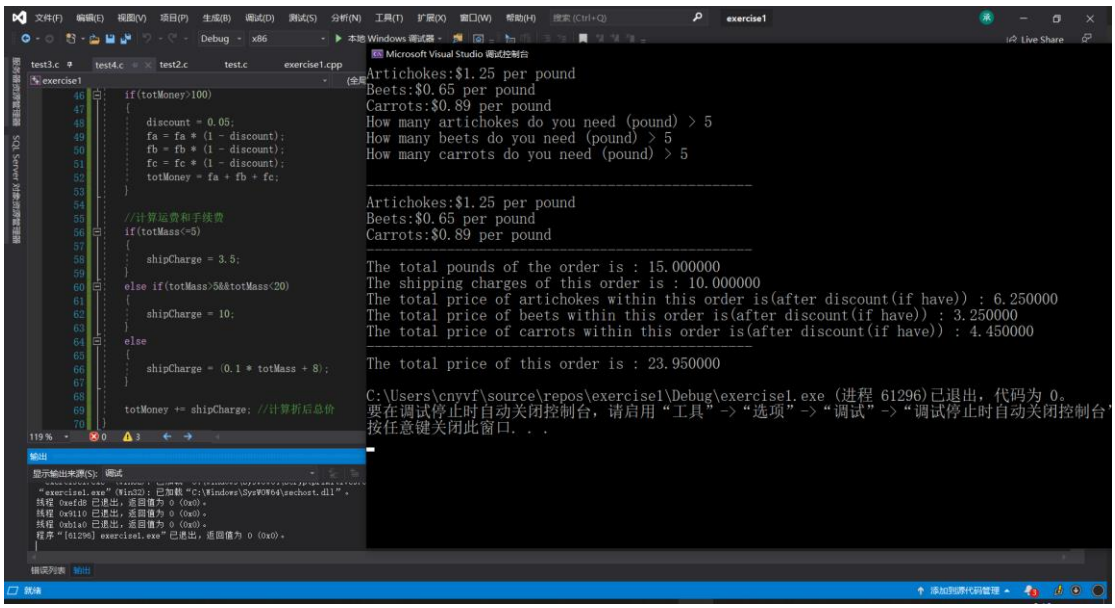
```
	fc = kc * mc;

	totMoney = fa + fb + fc;// Original total price

	// Calculate discount
	if(totMoney>100)
	{
		discount = 0.05;
		fa = fa * (1 - discount);
		fb = fb * (1 - discount);
		fc = fc * (1 - discount);
		totMoney = fa + fb + fc;
	}

	// Calculate shipping and handling charges
	if(totMass<=5)
	{
		shipCharge = 3.5;
	}
	else if(totMass>5&&totMass<20)
	{
		shipCharge = 10;
	}
	else
	{
		shipCharge = (0.1 * totMass + 8);
	}

	totMoney += shipCharge; // Calculate the total discounted price
}

// Show output results
void printfFinal()
{
	printfPerMoney();
	printf("-------------------------------------------------\n");
	if(discount!=0)	printf("The discount about this order is : %lf \n", discount);
	printf("The total pounds of the order is : %lf \n", totMass);
	printf("The shipping charges of this order is : %lf \n", shipCharge);
	printf("The total price of artichokes within this order is(after discount(if have)) : %lf \n",
fa);
	printf("The total price of beets within this order is(after discount(if have)) : %lf \n", fb);
	printf("The total price of carrots within this order is(after discount(if have)) : %lf \n", fc);
	printf("-------------------------------------------------\n");
```

```
    printf("The total price of this order is : %lf \n", totMoney);
}

void main()
{
    printfPerMoney();
    inputMass();
    calculateMoney();
    printf("\n");
    printf("-----------------------------------------------");
    printf("\n");
    printfFinal();

}
```

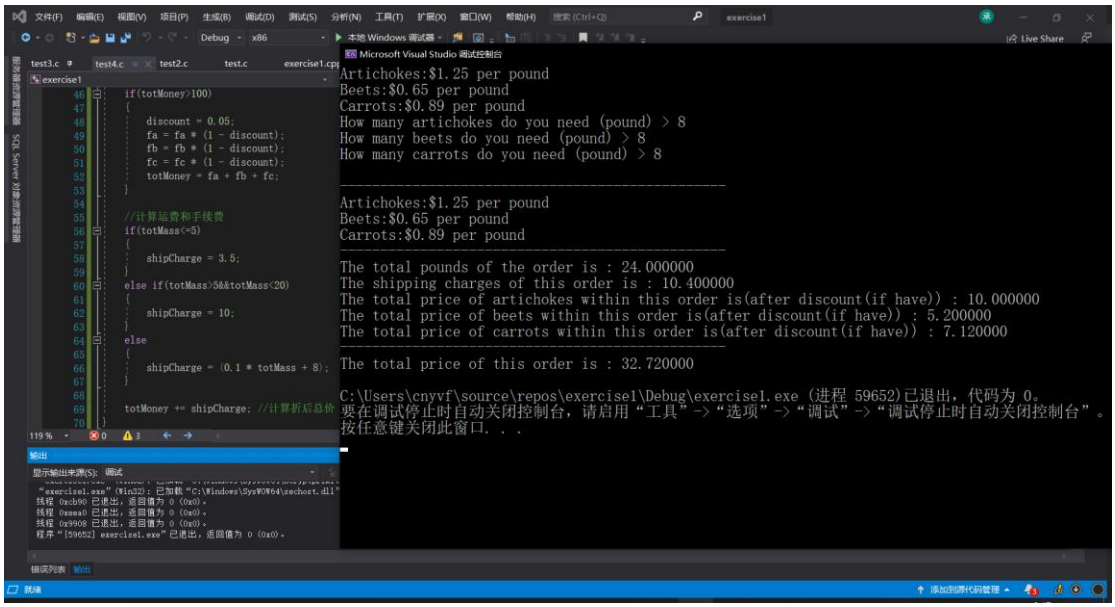
Case1: Total mass less than 5



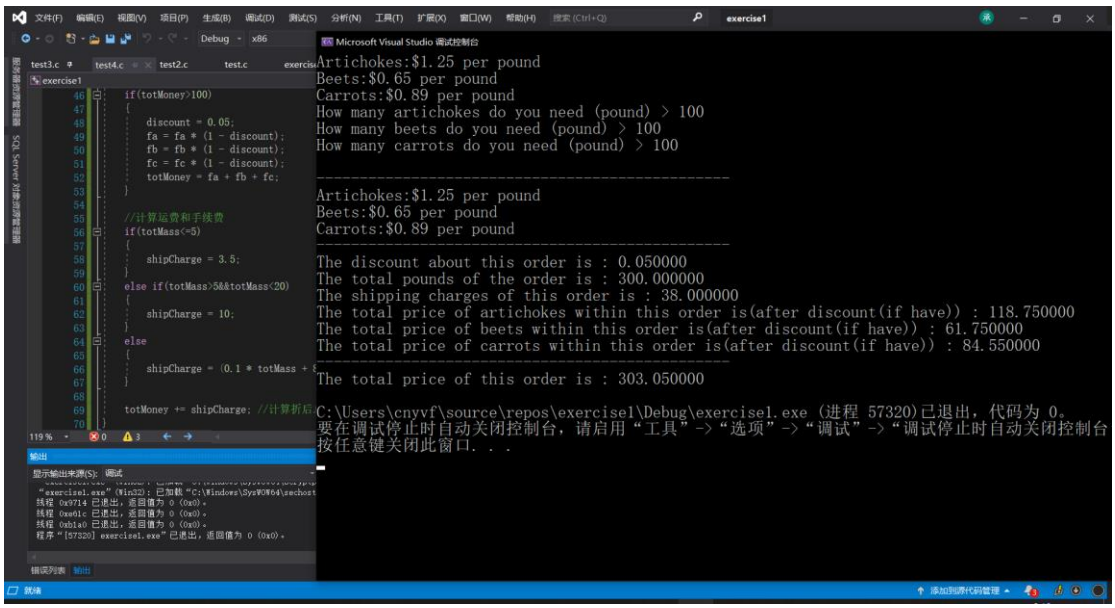
Case2: Total mass more than 5 and less than 20



Case3: Total mass more than 20



Case4: Total money more than 100


exercise1
Debug
x86
test3.c
test4.c
test2.c
test.c
exercise1
if(totMoney>100)
{
discount = 0.05;
fa = fa * (1 - discount);
fb = fb * (1 - discount);
fc = fc * (1 - discount);
totMoney = fa + fb + fc;
}
//计算运费和手续费
if(totMass<=5)
{
shipCharge = 3.5;
}
else if(totMass>5&&totMass<20)
{
shipCharge = 10;
}
else
{
shipCharge = (0.1 * totMass +
}
totMoney += shipCharge; //计算折后
}
119 %
输出
显示输出来源(S): 调试
“exercise1.exe”(Win32): 已加载 “C:\Windows\SysWOW64\sechost
线程 0x9714 已退出，返回值为 0 (0x0)。
线程 0xa01c 已退出，返回值为 0 (0x0)。
线程 0xb1a0 已退出，返回值为 0 (0x0)。
程序 “[57320] exercise1.exe” 已退出，返回值为 0 (0x0)。
错误列表
输出
就绪
添加到源代码管理
Microsoft Visual Studio 调试控制台
Artichokes:$1.25 per pound
Beets:$0.65 per pound
Carrots:$0.89 per pound
How many artichokes do you need (pound) > 100
How many beets do you need (pound) > 100
How many carrots do you need (pound) > 100
Artichokes:$1.25 per pound
Beets:$0.65 per pound
Carrots:$0.89 per pound
The discount about this order is : 0.050000
The total pounds of the order is : 300.000000
The shipping charges of this order is : 38.000000
The total price of artichokes within this order is(after discount(if have)) : 118.750000
The total price of beets within this order is(after discount(if have)) : 61.750000
The total price of carrots within this order is(after discount(if have)) : 84.550000
The total price of this order is : 303.050000
C:\Users\cnyvf\source\repos\exercise1\Debug\exercise1.exe (进程 57320)已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台
按任意键关闭此窗口. . .