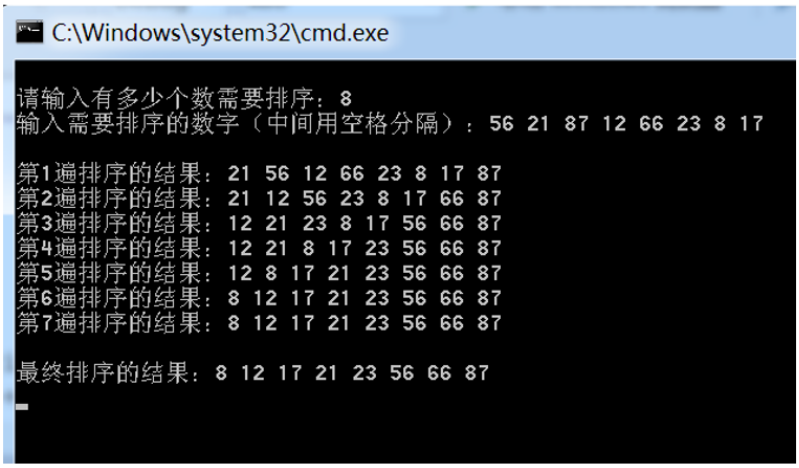
Exercise 1

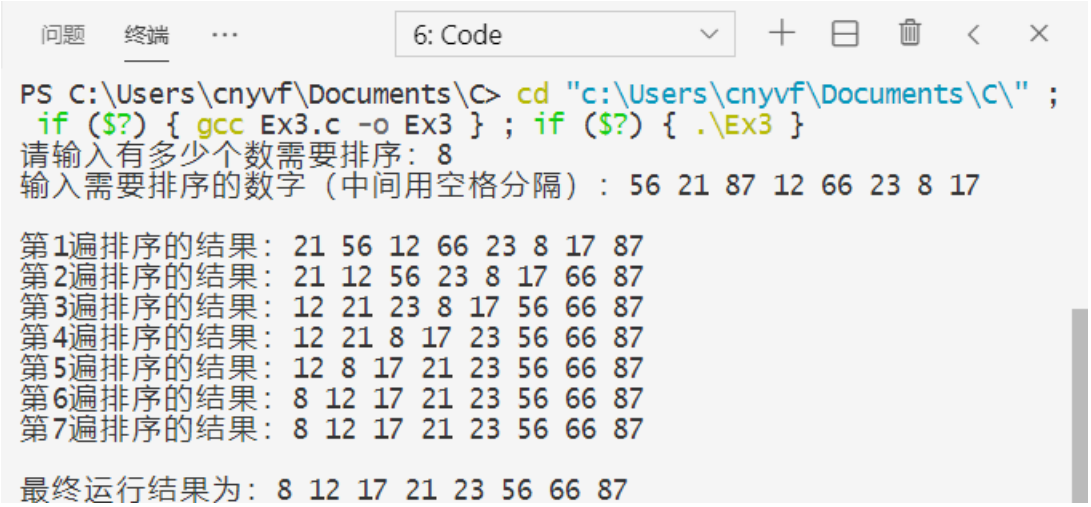
- Input any number of integers by the keyboard, and then output them in ascending order by the Bubble Method. You must also output the intermediate procedures of the sorting. The output result and requirement is shown in the following screenshot.




```c
#include <stdio.h>
int count; //一共有几个数
int nums[1000]; //存放数字用的数组

void bubbleSort();  //冒泡排序函数引用
void swap(int *a, int *b);  //元素换位函数引用

void main()
{
    printf("请输入有多少个数需要排序：");
    scanf("%d", &count);
    printf("输入需要排序的数字（中间用空格分隔）：");
    for(int i=0;i<count;i++)    //循环输入排序数字
    {
        scanf("%d", &nums[i]);
    }
    printf("\n");
    bubbleSort();
}

void bubbleSort()//冒泡排序函数
{
```

```
    int i;
    for(int j=0;j<count-1;j++)  //冒泡排序步骤
    {
        for (i = 0; i < count-1; i++)
        {
            if (nums[i] > nums[i + 1])
            {
                swap(&nums[i], &nums[i + 1]);
            }
        }
        printf("第%d 遍排序的结果：", j + 1);     //每排序完一遍输出结果
        for (i = 0; i < count; i++)
        {
            printf("%d ", nums[i]);
        }
        printf("\n");
    }
    printf("\n");
    printf("最终运行结果为：");
    for (i = 0; i < count; i++)
    {
        printf("%d ", nums[i]);
    }
}

void swap(int *a,int *b)    //元素换位函数
{
    int transTemp;
    transTemp = *a;
    *a = *b;
    *b = transTemp;
}
```

Case1:



Exercise 2

- Write a program 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, and put them into different arrays, and output the results by each category.

```
#include <stdio.h>
void main()
{
    char Input;
    int numLetters = 0, numSpaces = 0, numNumber = 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
    char letter[10000], space[10000], number[10000], other[10000];
    printf("Please input your string > ");
    while ((Input = 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("The string > ");
            pd = 1;
        }
        printf("%c", Input);//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 (Input >= 'A' && Input <= 'Z' || Input >= 'a' && Input <=
'z')
        {
            letter[numLetters] = Input;
            numLetters++;
        }
        else if (Input == ' ')
        {
            space[numSpaces] = Input;
            numSpaces++;
        }
        else if (Input >= '0' && Input <= '9')
        {
            number[numNumber] = Input;
            numNumber++;
        }
        else
        {
            other[numOther] = Input;
            numOther++;
        }
    }
    printf("\n");
    //输出结果
    printf("We have %d letters.  That are : ", numLetters);
    for(int i=0;i<numLetters;i++)
    {
        printf("%c ", letter[i]);
    }
    printf("\n");
    printf("We have %d spaces.  That are : ", numSpaces);
    for (int i = 0; i < numSpaces; i++)
    {
        printf("%c ", space[i]);
    }
    printf("\n");
    printf("We have %d numbers.  That are : ", numNumber);
    for (int i = 0; i < numNumber; i++)
    {
        printf("%c ", number[i]);
    }
    printf("\n");
```

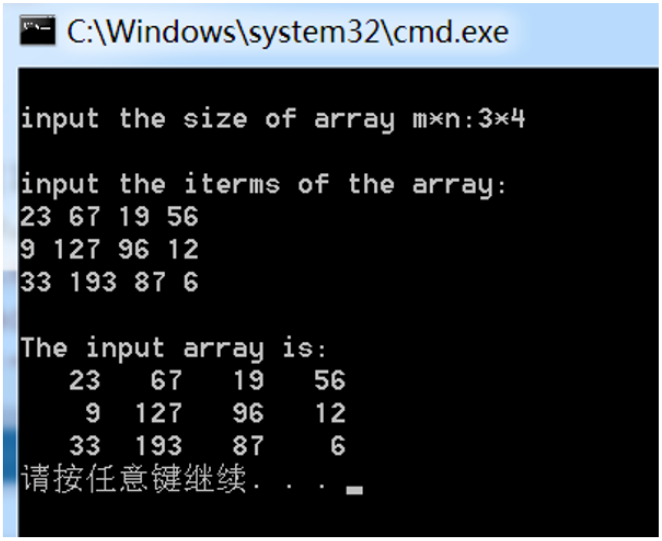
```
    printf("We have %d others.  That are : ", numOther);
    for (int i = 0; i < numOther; i++)
    {
        printf("%c ", other[i]);
    }
    printf("\n");
}
```

Case1:

```
问题  终端  ···                6: Code
PS C:\Users\cnyvf\Documents\C> cd "c:\Users\cnyvf\Documents\C\" ;
 if ($?) { gcc Ex3.c -o Ex3 } ; if ($?) { .\Ex3 }
Please input your string > AbcdE      1234567!@#$%^&*
The string > AbcdE      1234567!@#$%^&*
We have 5 letters.  That are : A b c d E
We have 6 spaces.  That are :
We have 7 numbers.  That are : 1 2 3 4 5 6 7
We have 8 others.  That are : ! @ # $ % ^ & *
PS C:\Users\cnyvf\Documents\C>
```

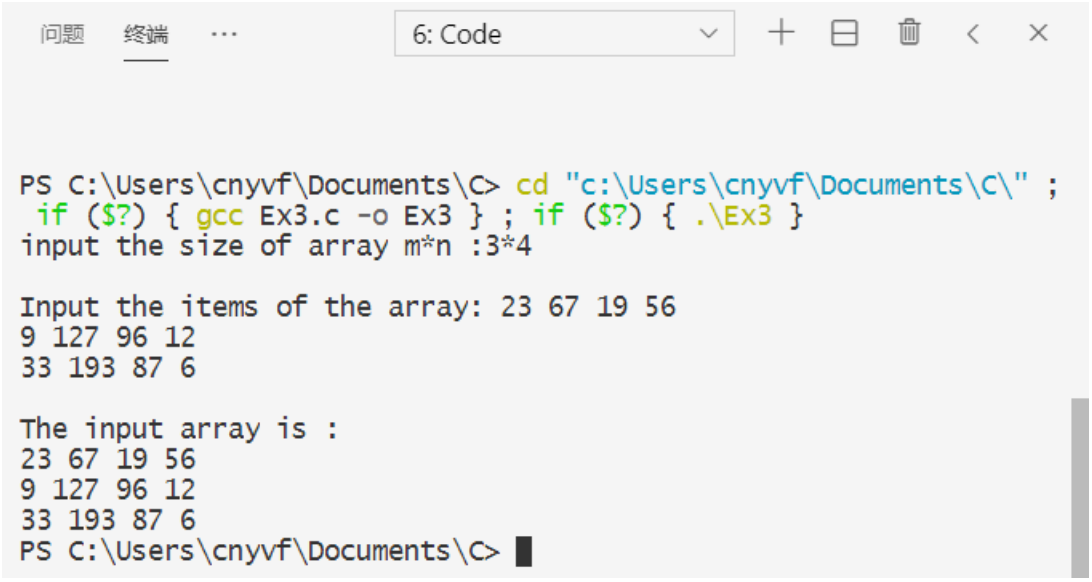
Exercise 3

- Input a two dimentional array by the keyboard, and then output this array on the screen, the required format is shown in the following picture.



```c
#include <stdio.h>

void main()
{
    int m, n; //用于定义二维数组大小
    int massMent[100][100]; //定义二维数组
    printf("input the size of array m*n :");
    scanf("%d*%d", &m, &n);
    printf("\n");
    printf("Input the items of the array: ");
    //输入数组值
    for(int i=0;i<m;i++){
        for(int j=0;j<n;j++){
            scanf("%d",&massMent[i][j]);
        }
    }
    printf("\n");
    printf("The input array is : \n");
    //输出数组值
    for(int i=0;i<m;i++){
        for(int j=0;j<n;j++){
            printf("%d ",massMent[i][j]);
        }
        printf("\n");
    }
}
```

Case1:



Exercise 4

- Write a program to connect two strings, and put the second string after the first string. The two strings are input from the keyboard and they are stored in two character arrays. Do not use the third array and do NOT use the strcat() function.

```c
#include <stdio.h>
void main()
{
    char str_1[10000],str_2[5000];  //定义两个字符数组用于储存字符串
    char temp;  //缓存输入值
    int num1=0,num2=0;  //记录填充位数
    printf("Please input the str_1 : ");
    while((temp=getchar())!='\n'){  //输入字符串 1
        str_1[num1]=temp;
        num1++;
    }
    printf("Please input the str_2 : ");
    while((temp=getchar())!='\n'){  //输入字符串 2
        str_2[num2]=temp;
        num2++;
    }
    for(int i=num1;i<=num1+num2;i++){    //拼接字符串 1、2
        str_1[i]=str_2[i-num1];
    }
    printf("The string is : %s",str_1); //输出
}
```

Case1:

```
问题  终端  ···   6: Code
PS C:\Users\cnyvf\Documents\C> cd "c:\Users\cnyvf\Documents\C\" ;
 if ($?) { gcc Ex3.c -o Ex3 } ; if ($?) { .\Ex3 }
Please input the str_1 : asdfghjkl:"
Please input the str_2 : qwertyuiop{}
The string is : asdfghjkl:"qwertyuiop{}
PS C:\Users\cnyvf\Documents\C>
```

Exercise 5

- Find a saddle point for a two dimentional array，the saddle point is the maximum element in its row, and the minimum element in its column. The two dimentional array and its size should be input from the keyboard. Notice that the saddle point may not exist for an array, if it does not exist, output the information that the saddle point does not exist. If there exists a saddle point, please output its row number and column number.

```
input row size(number of columns): 3
input column size(number of rows): 3

input the iterms of row0: 3   9   12
input the iterms of row1: 2   7   8
input the iterms of row2: 16  4   13

The array you input is:
3   9   12
2   7   8
16  4   13

The saddle point is found! Its position is: array[1][2], and array[1][2]=8
```

```c
#include <stdio.h>
int MassMent[100][100];
int maxR[100]; //用于储存每一行的最大值的列地址
void main()
{
    int r,c;    //定义数组的长和宽
    int temp=0; //用于临时存储数据
    printf("input row size(number of columns): ");
    scanf("%d",&r);
    printf("input column size(number of rows): ");
    scanf("%d",&c);
    for(int i=0;i<r;i++){   //输入二维数组的元素，并找出每一行最大值的地址
        printf("input the items of row%d: ",i+1);
        for(int j=0;j<c;j++){
            scanf("%d",&MassMent[i][j]);
            if(MassMent[i][j]>temp) {
                temp=MassMent[i][j];
                maxR[i]=j;
            }
```

```
            }
            temp=0;
        }
        printf("\nThe array you input is: \n");
        for(int i=0;i<r;i++){   //输出二维数组
            for(int j=0;j<c;j++){
                printf("%d ",MassMent[i][j]);
            }
            printf("\n");
        }
        printf("\n");
        int text=0;
        for(int i=0;i<r;i++){   //根据每一行的最大值的位置进行列遍历，找出鞍
点
            int pd=1;
            for(int j=0;j<r;j++){
                if(MassMent[i][maxR[i]]>MassMent[j][maxR[i]]){
                    pd=0;
                }
            }
            if(pd==1&&text==0){
                printf("The saddle point is found! Its position is :");
                text=1;
            }
            if(pd==1){
                printf("Array[%d][%d], and Array[%d][%d]=%d;",i,maxR[i],
i,maxR[i],MassMent[i][maxR[i]]);
            }
        }
        if(text==0){    //若无鞍点 输出错误提示
            printf("The saddle point is not found!");
        }
}
```

Case1:

```
问题  输出  调试控制台  终端        2: Code

C:\Users\YH\Documents\C++>cd "c:\Users\YH\Documents\C++\" && gcc Exercise_3.c
 -o Exercise_3 && "c:\Users\YH\Documents\C++\"Exercise_3
input row size(number of columns): 3
input column size(number of rows): 3
input the items of row1: 3 9 12
input the items of row2: 2 7 8
input the items of row3: 16 4 13

The array you input is:
3 9 12
2 7 8
16 4 13

The saddle point is found! Its position is :Array[1][2], and Array[1][2]=8;
C:\Users\YH\Documents\C++>
```

Exercise 6

There is a m*n two-dimentional array, where m and n are the sizes of the array and they could be any numbers that you can change their values. Find the maximum five elements of the array, and output the values of the five maximum elements and their positions. As the following examples:

```
The input array is:
   23   42  234  342   13
   20   17  268  984   36
   12  486   96  239   11
  456  237  634  124    2
  131   54  324   23   99

The maximum 1th number: a[1][3]=984
The maximum 2th number: a[3][2]=634
The maximum 3th number: a[2][1]=486
The maximum 4th number: a[3][0]=456
The maximum 5th number: a[0][3]=342
```

```
The input array is:
   23   42  234  342   13
   20   17  268  984   36
   12  486   96  239   11

The maximum 1th number: a[1][3]=984
The maximum 2th number: a[2][1]=486
The maximum 3th number: a[0][3]=342
The maximum 4th number: a[1][2]=268
The maximum 5th number: a[2][3]=239
```

```c
#include <stdio.h>
#include <stdlib.h>
int massMent[100][100]; //二维数组
int maxNums[5]; //记录前五最大值
int saveI[5],saveJ[5];  //记录最大值对应的坐标
int smallpoint=0;   //记录最大值中的最小值
int m,n;    //横纵大小
struct maxPoint{    //结构体用于绑定最大值及其坐标，便于排序
    int num;
```

```
    int ix;
    int jx;
}mp[5];
int cmp(const void*a,const void*b)  //比较函数,通过比较结构体中的 num 来
逆序排序
{
    struct maxPoint xx=*(struct maxPoint*)a;
    struct maxPoint yy=*(struct maxPoint*)b;
    return yy.num-xx.num;
}
void main()
{
    printf("input row size: ");
    scanf("%d",&m);
    printf("input column size: ");
    scanf("%d",&n);
    printf("Please input the array: \n");
    for(int i=0;i<m;i++){
        for(int j=0;j<n;j++){
            scanf("%d",&massMent[i][j]);    //输入二维数组
        }
    }
    printf("\nThe input array is: \n");
    for(int i=0;i<m;i++){
        for(int j=0;j<n;j++){
            printf("%d ",massMent[i][j]);
            if(massMent[i][j]>maxNums[smallpoint]){ //搜索二维数组并获
取前五最大值
                maxNums[smallpoint]=massMent[i][j];
                saveI[smallpoint]=i;
                saveJ[smallpoint]=j;
            }
            for(int x=0;x<5;x++){   //刷新前五最大值中的最小值点
                if(maxNums[x]<maxNums[smallpoint]){
                    smallpoint=x;
                }
            }
        }
        printf("\n");
    }
    printf("\n");
    for(int i=0;i<5;i++){   //将已有的数值导入结构体
        mp[i].num=maxNums[i];
        mp[i].ix=saveI[i];
```

```
        mp[i].jx=saveJ[i];
    }
    qsort(mp,5,sizeof(mp[0]),cmp);  //结构体排序
    for(int i=0;i<5;i++){   //输出结果
        printf("The maximum %dth number: a[%d][%d]=%d \n",i+1,mp[i].
ix,mp[i].jx,mp[i].num);
    }
}
```

Case1: