Experiment 6 Array

1. Experimental Purpose

(1) Master the definition, assignment, input and output methods of one-dimensional and two-dimensional arrays.

(2) Master the use of character arrays and string functions.

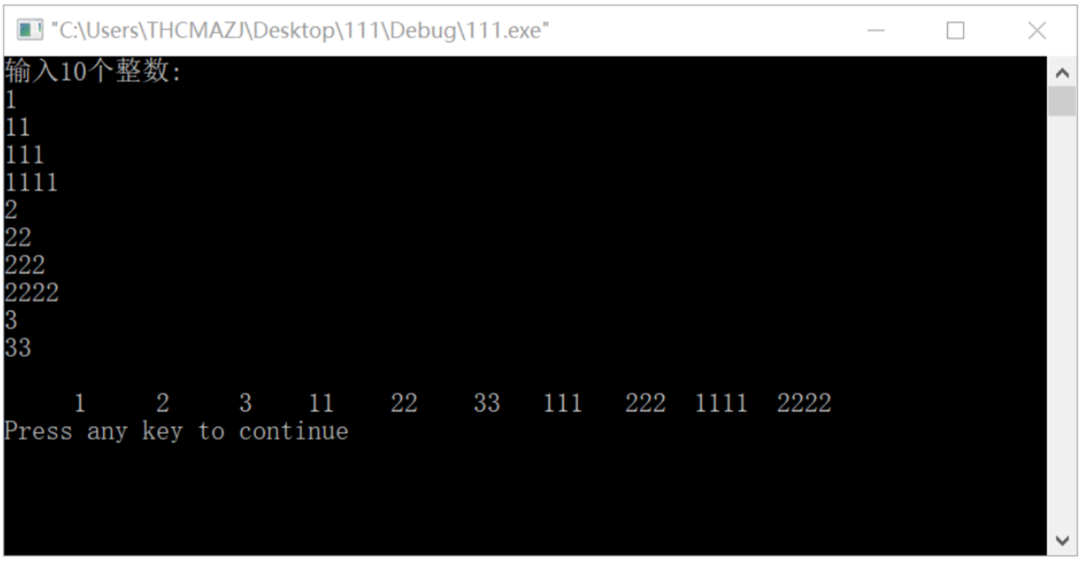
(3) Master algorithms related to arrays (especially sorting algorithms).

2. Experimental Contents

(1) Use selection to sort 10 integers. Ten integers are entered using the scanf function.

Code:

```
#include <stdio.h>
int main()
{
    int i,j,t;
    int s[10];
    printf("Enter 10 integers:\n");
    for (i=0;i<10;i++)
        scanf("%d",&s[i]);
    for (i=0;i<10;i++)
        for (j=0;j<9-i;j++)
            if(s[j]>s[j+1])
            {
                t=s[j];
                s[j]=s[j+1];
                s[j+1]=t;
            }
    printf("\n");
    for (i=0;i<10;i++)
        printf("%6d",s[i]);
    printf("\n");
    return 0;
}
```



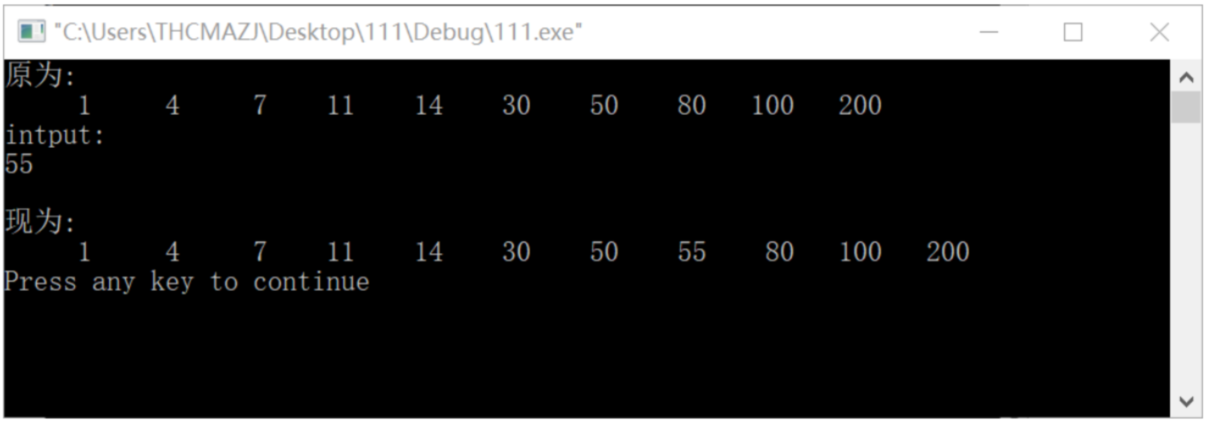
```
输入10个整数:
1
11
111
1111
2
22
222
2222
3
33

     1     2     3    11    22    33   111   222  1111  2222
Press any key to continue
```

(2) There is already an array that has been sorted. After a number has been entered, it is inserted into the array according to the original sorting rules.

Code:

```
#include <stdio.h>
int main()
{
    int n,i,j,t;
    int a[12]={0,1,4,7,11,14,30,50,80,100,200};
    printf("原为:\n");
    for (i=1;i<11;i++)
        printf("%6.d",a[i]);
    printf("\n");
    printf("intput:\n");
    scanf("%d",&n);
    a[11]=n;
        for (i=11;i>1;i--)
            if (a[i]<a[i-1])
            {
                t=a[i];
                a[i]=a[i-1];
                a[i-1]=t;
            }
    printf("\n is now: \n");
    for (i=1;i<=11;i++)
        printf("%6.d",a[i]);
    printf("\n");
    return 0;
}
```



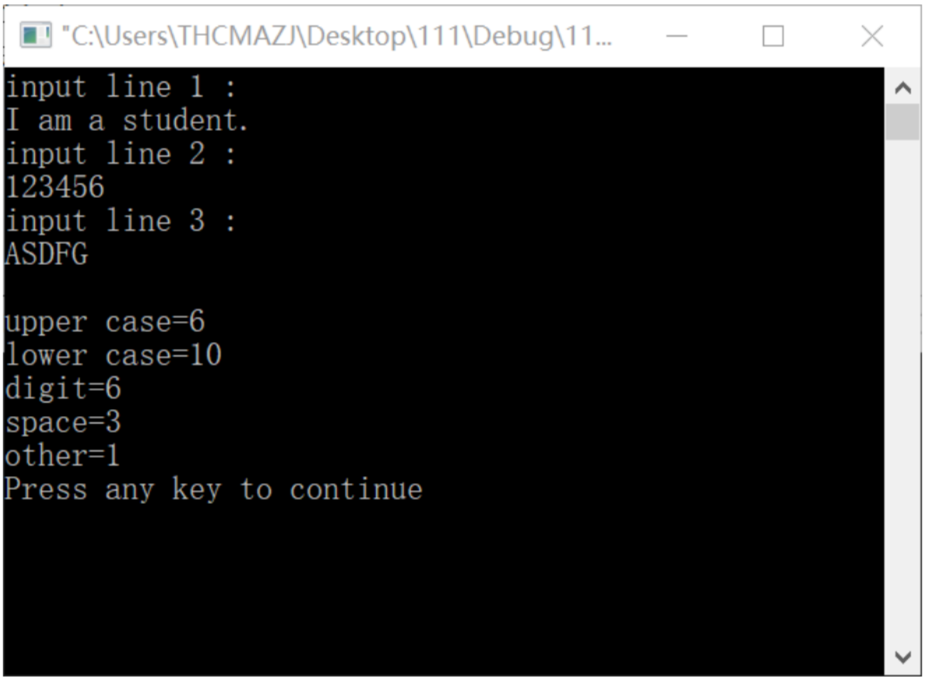
```
原为:
     1     4     7    11    14    30    50    80   100   200
intput:
55

现为:
     1     4     7    11    14    30    50    55    80   100   200
Press any key to continue
```

(3)There is an article with three lines of text, each with 80 characters. Requires that the number of English uppercase letters, lowercase letters, numbers, spaces and other characters be counted separately.

Code:

```
#include <stdio.h>
int main()
{
    char a[3][80];
    int i,j,upp,low,dig,spa,oth;
    upp=low=oth=spa=dig=0;
    for (i=0;i<3;i++)
    {
        printf("input line %d :\n",i+1);
        gets(a[i]);
        for(j=0;j<80&&a[i][j]!='\0';j++)
        {
            if(a[i][j]>='A'&&a[i][j]<='Z')
                upp++;
            else if(a[i][j]>='a'&&a[i][j]<='z')
                low++;
            else if(a[i][j]>='0'&&a[i][j]<='9')
                dig++;
            else if(a[i][j]==' ')
                spa++;
            else
                oth++;
        }
    }
    printf("\nupper case=%d\nlower
case=%d\ndigit=%d\nspace=%d\nother=%d\n",upp,low,dig,spa,oth);
    return 0;
}
```



```
input line 1 :
I am a student.
input line 2 :
123456
input line 3 :
ASDFG

upper case=6
lower case=10
digit=6
space=3
other=1
Press any key to continue
```

(4)Find the saddle point of a two-dimensional array, that is, the element at that location is the largest in the row, the smallest in the column, or there may be no saddle point.

At least two sets of test data should be prepared:

①Two-dimensional arrays have saddle points, for example:

9	80	205	40
90	-60	96	1
210	-3	101	89

②Two-dimensional arrays have no saddle points, for example:

9	80	205	40
90	-60	196	1
210	-3	101	89
45	54	156	7

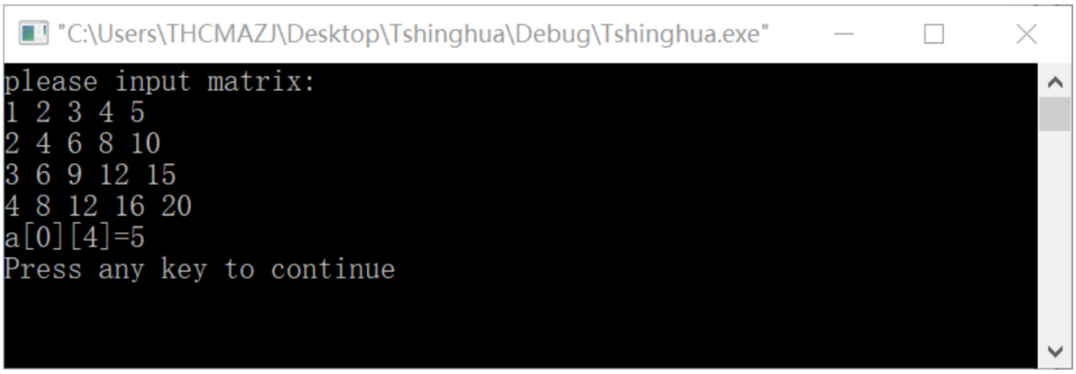
Use the scanf function to enter values for each element of the array from the keyboard and check that the results are correct. The title does not specify the number of rows and columns in a two-dimensional array. The program should be able to handle arrays of any number of rows and columns. Therefore, in theory, you should prepare many different types of array data with different numbers of rows and columns, but this is too much work and generally you don't need to prepare only typical data.

If you have specified the number of rows and columns in the array, you can initialize the array elements in your program without using the scanf function.

Code:

```
#include <stdio.h>
#define N 4
#define M 5
```

```
int main()
{
    int i,j,k,a[N][M],max,maxj,flag;
    printf("please input matrix:\n");
    for (i=0;i<N;i++)
        for (j=0;j<M;j++)
            scanf("%d",&a[i][j]);
    for (i=0;i<N;i++)
    {
            max=a[i][0];
            maxj=0;
        for (j=0;j<M;j++)
                if(a[i][j]>max)
                {
                    max=a[i][j];
                    maxj=j;
                }
        flag=1;
        for (k=0;k<N;k++)
            if(max>a[k][maxj])
            {
                flag=0;
                continue;
            }
        if(flag)
        {
            printf("a[%d][%d]=%d\n",i,maxj,max);
            break;
        }
    }
    if(!flag)
        printf("It is not exist!\n");
    return 0;
}
```



```
please input matrix:
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
a[0][4]=5
Press any key to continue
```