**实验报告**

|  |  |
| --- | --- |
| **课程名称** | **程序设计基础（C语言）** |
| **实验项目** | **实验四 二维数组** |
|  |  |

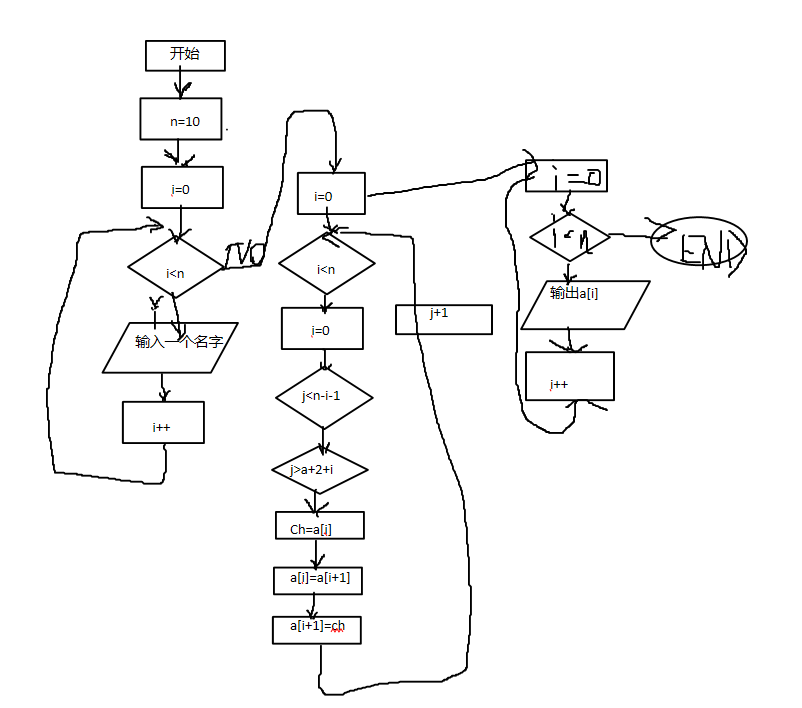
|  |  |  |
| --- | --- | --- |
| **学院** |  | **计算机学院** |
| **系别** |  | **计算机大类** |
| **班级/学号** |  | **计类1607/2016011353** |
| **学生姓名** |  | **郭泽晨** |
| **实验日期** |  | **2016.11.09** |
| **成绩** |  |  |
| **指导教师** |  | **蒋玉茹** |

## 实验四 二维数组

1. **实验目的**
2. 熟练掌握三种循环语句的应用。
3. 熟练掌握循环结构的嵌套。
4. 练习调试与修改程序。
5. **实验内容**

**题目1：输入十个人的姓名，并按升序输出。（用选择法或者冒泡法排序）**

**流程图：**



**程序：**

**#include<stdio.h>**

**#include<string.h>**

**#define n 10**

**int main()**

**{**

**char a[n][20];**

**char ch[20];**

**int i,j;**

**int t;**

**printf("请输入十个名字，按回车输入下一个名字:\n");**

**for(i=0;i<n;i++)**

**{**

**gets(a[i]);**

**}**

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<n-i-1;j++)**

**{**

**if(strcmp(a[j],a[j+1])>0)**

**{**

**strcpy(ch,a[j]);**

**strcpy(a[j],a[j+1]);**

**strcpy(a[j+1],ch);**

**}**

**}**

**}**

**printf("升序的结果:\n");**

**for(i=0;i<n;i++)**

**{**

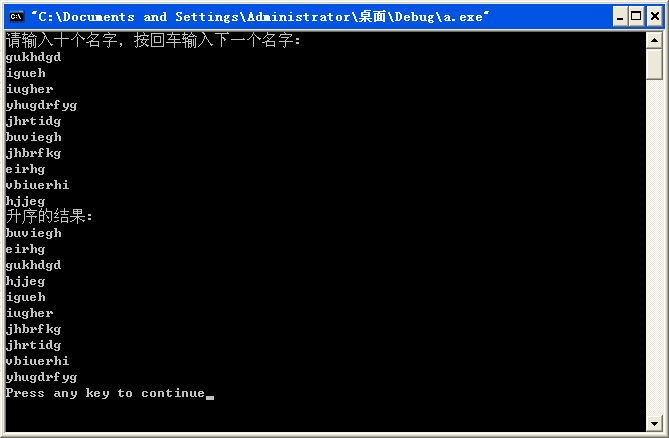
**puts(a[i]);**

**}**

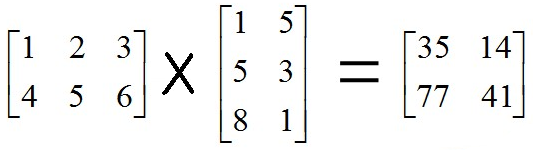
**return 0;**

**}**

**结果:**

****

**题目2：求两个矩阵的积。如：**



运算规则为：

第i行第j列相乘得到结果矩阵中r[i][j]的值。比如：

r[0][0] = 1\*1+2\*5+3\*8=35;

r[0][1] = 1\*5+2\*3+3\*1=14;

r[1][0] = 4\*1+5\*5+6\*8=77;

r[1][1] = 4\*5+5\*3+6\*1=41;

要求：

1.数组的行数和列数不超过100。

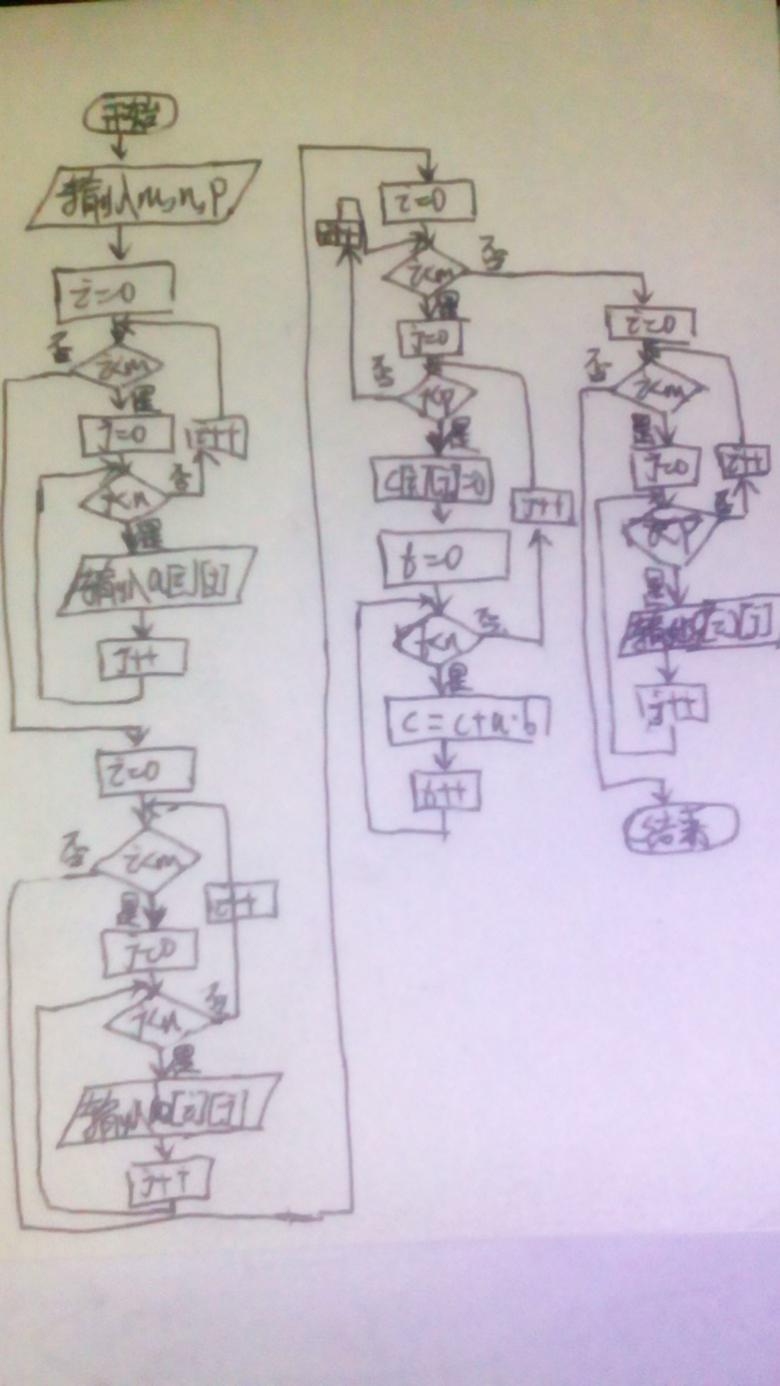
2.输入第一行中，第一个正整数m，表示第一个矩阵的行数，输入第二个正整数n，表示第一个矩阵的列数，也即是第二个矩阵的行数，输入第三个正整数p，表示第二个矩阵的列数。

3.接下来的m\*n个数是第一个矩阵中的数。

4.再接下来的n\*p个数是第二个矩阵中的数。

5.输出计算结果，为一个m行，p列的矩阵。

流程图：



程序：

#include<stdio.h>

int main()

{

int a[100][100];

int b[100][100];

int c[100][100];

int m,n,p;

int i,j;

int t;

printf("输入第一个矩阵的行m、列n，第二个矩阵的列p:");

scanf("%d%d%d",&m,&n,&p);

for(i=0;i<m;i++)

{

for(j=0;j<n;j++)

{

scanf("%d",&a[i][j]);

}

}

for(i=0;i<n;i++)

{

for(j=0;j<p;j++)

{

scanf("%d",&b[i][j]);

}

}

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

c[i][j]=0;

for(t=0;t<n;t++)

{

c[i][j]=c[i][j]+a[i][t]\*b[t][j];

}

}

}

for(i=0;i<m;i++)

{

for(j=0;j<p;j++)

{

printf("%d ",c[i][j]);

}

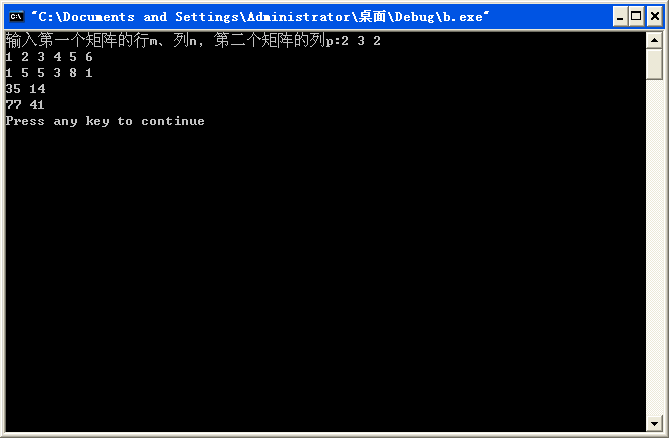
printf("\n");

}

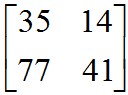
return 0;

}

结果：



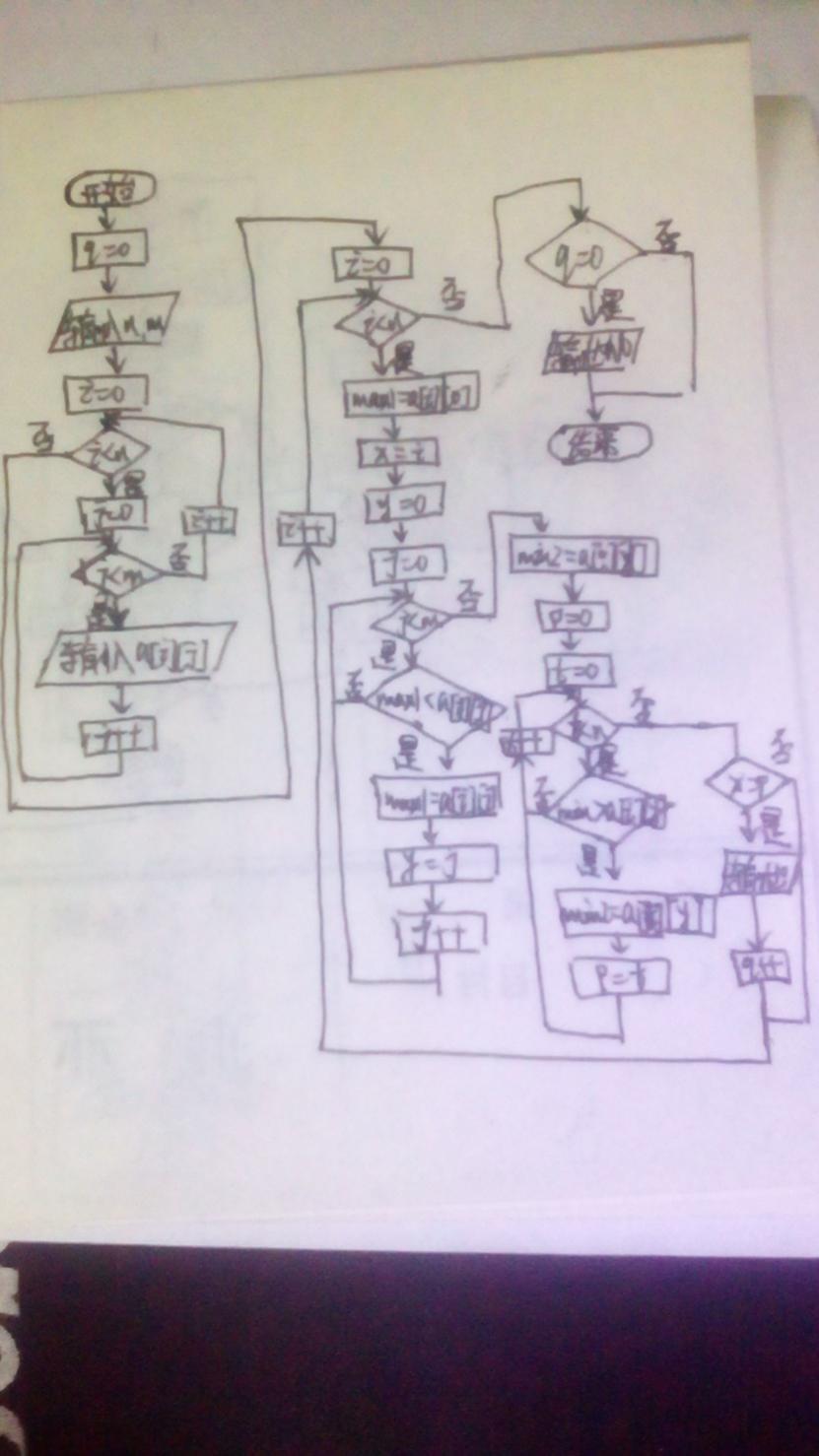
**题目3：求矩阵的鞍点。鞍点位置上的元素在该行中值最大，在该列中值最小。如下面矩阵的鞍点为35。**



【输入】前两个数为行数和列数，其后是矩阵中的元素

【输出】如果没有鞍点，输出NO；如果有输出格式为：[值1，行下标1，列下标1]， [值2，行下标2，列下标2]……

流程图：



程序：

#include<stdio.h>int main()

{

int n,m;

int i,j;

int t;

int p,q=0;

int x,y;

int max1,min2;

int a[10][10];

printf("输入行数和列数n,m:");

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

for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

{

scanf("%d",&a[i][j]);

}

}

for(i=0;i<n;i++)

{

max1=a[i][0];

x=i;

y=0;

for(j=0;j<m;j++)

{

if(max1<=a[i][j])

{

max1=a[i][j];

y=j;

}

}

min2=a[0][y];

p=0;

for(t=0;t<n;t++)

{

if(min2>=a[t][y])

{

min2=a[t][y];

p=t;

}

}

if(x==p)

{

printf("[%d,%d,%d] ",a[x][y],x,y);

q++;

}

}

if(q==0)

{

printf("NO");

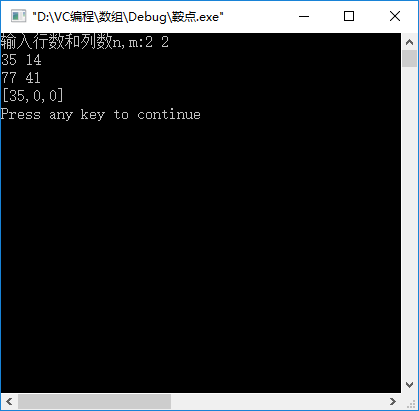
}

printf("\n");

return 0;

}

结果：



**题目4：井字格游戏**

**程序大致的执行过程如图所示：**

功能包括：

1.允许两个玩家下棋

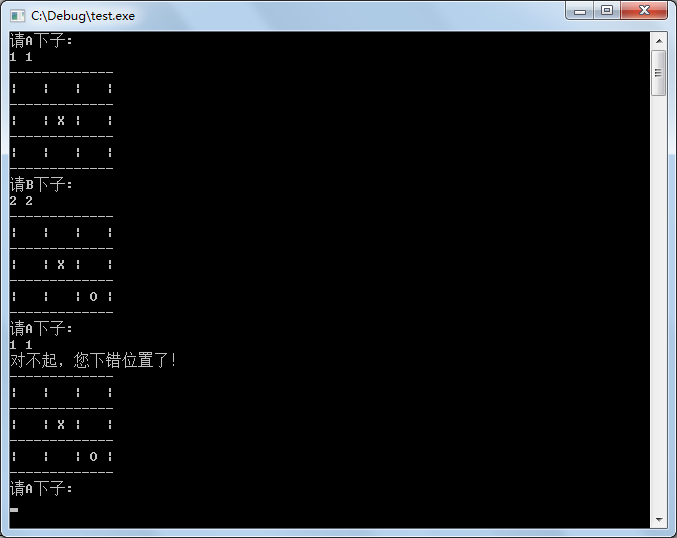
2.允许玩家输入下棋的坐标

3.显示当前棋盘状态

4.判断玩家是否下错位置，即在已经下过棋的位置再次下棋

5.判断玩家下棋后，输赢状态

6.在屏幕提示中指导玩家交替下棋



1. **实验心得**

1.for语句书写中不能随便加分号

比如:

for();

{}

其中的分号是不应该添加的。

2.取一个数中各个位的数字的方法。

综合运用了整除和求余两个符号。

比如：

12%10得到个位数字2

12/10得到十位数字1