

C语言刷题训练营-第十讲

比特就业课@提供课程讲解和技术支持

题目来自: [牛客网](#)

BC101-班级成绩输入输出

题目描述

输入一个班级5个学生各5科成绩，输出5个学生各5科成绩及总分。

输入描述:

五行，每行输入一个学生各5科成绩（浮点数表示，范围0.0~100.0），用空格分隔。

输出描述:

五行，按照输入顺序每行输出一个学生的5科成绩及总分（小数点保留1位），用空格分隔。

示例1

输入

```
98.5 96.0 100 99 88
60.5 77 88.5 99 60
80 80.5 89 91 92
93 94 95.5 96 97.5
100 98 88.5 88 78.5
```

输出

```
98.5 96.0 100.0 99.0 88.0 481.5
60.5 77.0 88.5 99.0 60.0 385.0
80.0 80.5 89.0 91.0 92.0 432.5
93.0 94.0 95.5 96.0 97.5 476.0
100.0 98.0 88.5 88.0 78.5 453.0
```

参考代码:

```
#include <stdio.h>

int main()
{
    int i = 0;

    //5个学生
    for(i=0; i<5; i++)
    {
        double sum = 0.0;
```

```
int j = 0;
double score = 0.0;
for(j=0; j<5; j++)
{
    scanf("%lf", &score);
    sum += score;
    printf("%.1lf ", score);
}
printf("%.1lf\n", sum);
}
return 0;
}
```

BC102-矩阵元素定位

题目描述

Kiki得到了一个n行m列的矩阵，现在他想知道第x行第y列的值是多少，请你帮助他完成这个任务。

输入描述:

第一行包含两个数n和m，表示这个矩阵包含n行m列。从2到n+1行，每行输入m个整数（范围-231~231-1），用空格分隔，共输入n*m个数，表示矩阵中的元素。接下来一行输入x和y，用空格分隔，表示Kiki想得到的元素的位置。（ $1 \leq x \leq n \leq 10$, $1 \leq y \leq m \leq 10$ ）

输出描述:

一行，输出一个整数值，为Kiki想知道的值。

示例1

输入

```
2 3
1 2 3
4 5 6
1 2
```

输出

```
2
```

参考代码:

```
#include <stdio.h>

int main()
{
    int n = 0;
    int m = 0;
    int x = 0;
    int y = 0;
```

```
scanf("%d %d", &n, &m);
//这是c99语法-变长数组，但是数组不能初始化
int arr[n][m]; //具体给值也行
int i = 0;
int j = 0;
for(i=0; i<n; i++)
{
    for(j=0; j<m; j++)
    {
        scanf("%d", &arr[i][j]);
    }
}

scanf("%d%d", &x, &y);
printf("%d\n", arr[x-1][y-1]);
return 0;
}
```

BC103-序列重组矩阵

题目描述

KiKi现在得到一个包含 nm 个数的整数序列，现在他需要把这 nm 个数按顺序规划成一个 n 行 m 列的矩阵并输出，请你帮他完成这个任务。

输入描述:

一行，输入两个整数 n 和 m ，用空格分隔，第二行包含 $n*m$ 个整数（范围 $-231 \sim 231-1$ ）。（ $1 \leq n \leq 10$ ， $1 \leq m \leq 10$ ）

输出描述:

输出规划后 n 行 m 列的矩阵，每个数的后面有一个空格。

示例1

输入

```
2 3
1 2 3 4 5 6
```

输出

```
1 2 3
4 5 6
```

参考代码:

```
#include <stdio.h>

int main()
{
```

比特就业课，400小时就业课，保姆式就业服务

```
int n = 0;
int m = 0;
scanf("%d%d", &n, &m);
int i = 0;
int tmp = 0;
for(i=1; i<=m*n; i++)
{
    scanf("%d", &tmp);
    printf("%d ", tmp);
    if(i%m == 0)
        printf("\n");
}
return 0;
}
```

BC104-最高身高

题目描述

KiKi想从n行m列的方阵队列中找到身高最高的人的位置，请你帮助他完成这个任务。

输入描述:

第一行包含两个整数n和m，表示这个方阵队列包含n行m列。从2到n+1行，每行输入m个整数（范围-231~231-1），用空格分隔，共输入n*m个数，表示方阵中的所有人的身高（保证输入身高都不相同）。(1≤x≤n≤10, 1≤y≤m≤10)

输出描述:

一行，输出两个整数，用空格分隔，表示方阵中身高最高的人所在的行号和列号。

示例1

输入

```
2 2
175 180
176 185
```

输出

```
2 2
```

参考代码:

```
int main()
{
    int n = 0;
    int m = 0;
    scanf("%d %d", &n, &m);
    int i = 0;
    int j = 0;
```

```

int max = 0; 比特就业课，400小时就业课，保姆式就业服务
int x = 0;
int y = 0;
for(i=0; i<n; i++)
{
    for(j=0; j<m; j++)
    {
        int hi = 0;
        scanf("%d", &hi);
        if(hi>max)
        {
            x = i;
            y = j;
            max = hi;
        }
    }
}
printf("%d %d\n", x+1,y+1);
return 0;
}

```

BC105-矩阵相等判断

题目描述

KiKi得到了两个 n 行 m 列的矩阵，他想知道两个矩阵是否相等，请你回答他。(当两个矩阵对应数组元素都相等时两个矩阵相等)。

输入描述:

第一行包含两个整数 n 和 m ，表示两个矩阵包含 n 行 m 列，用空格分隔。从2到 $n+1$ 行，每行输入 m 个整数（范围 $-231 \sim 231-1$ ），用空格分隔，共输入 $n*m$ 个数，表示第一个矩阵中的元素。
从 $n+2$ 行到 $2n+1$ ，每行输入 m 个整数（范围 $-231 \sim 231-1$ ），用空格分隔，共输入 $n*m$ 个数，表示第二个矩阵中的元素。
 $1 < n, m < 10$

输出描述:

一行，如果两个矩阵相等输出"Yes"并换行，否则输出"No"并换行。

示例1

输入

```

2 2
1 2
3 4
1 2
3 4

```

输出

```

Yes

```

```
#include <stdio.h>

int main()
{
    int n = 0;
    int m = 0;
    scanf("%d%d", &n, &m);
    int a1[n][m];
    int a2[n][m];
    int i = 0;
    int j = 0;
    for(i=0; i<n; i++)
    {
        for(j=0; j<m; j++)
        {
            scanf("%d", &a1[i][j]);
        }
    }
    for(i=0; i<n; i++)
    {
        for(j=0; j<m; j++)
        {
            scanf("%d", &a2[i][j]);
        }
    }
    for(i=0; i<n; i++)
    {
        for(j=0; j<m; j++)
        {
            if(a1[i][j] != a2[i][j])
                break;
        }
    }
    if(i==n && j==m)
    {
        printf("Yes\n");
    }
    else
    {
        printf("No\n");
    }
    return 0;
}
```

BC106-上三角矩阵判断

题目描述

KiKi想知道一个n阶方矩是否为上三角矩阵, 请帮他编程判定。上三角矩阵即主对角线以下的元素都为0的矩阵, 主对角线为从矩阵的左上角至右下角的连线。

输入描述:

第一行包含一个整数 n ，表示一个方阵包含 n 行 n 列，用空格分隔。（ $1 \leq n \leq 10$ ）从2到 $n+1$ 行，每行输入 n 个整数（范围 $-231 \sim 231-1$ ），用空格分隔，共输入 $n*n$ 个数。

输出描述:

一行，如果输入方阵是上三角矩阵输出"YES"并换行，否则输出"NO"并换行。

示例1

输入

```
3
1 2 3
0 4 5
0 0 6
```

输出

YES

参考代码:

```
#include <stdio.h>

int main()
{
    int n = 0;
    scanf("%d", &n);
    int a[n][n];
    int i = 0;
    int j = 0;
    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    //判断
    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)
        {
            if(i>j)
            {
                if(a[i][j] != 0)
                {
                    //下三角有不是0的就不满足
                    printf("NO");
                    return 0;
                }
            }
        }
    }
    printf("YES");
}
```

```
return 0;    比特就业课，400小时就业课，保姆式就业服务
}
```

BC107-矩阵转置

题目描述

KiKi有一个矩阵，他想知道转置后的矩阵（将矩阵的行列互换得到的新矩阵称为转置矩阵），请编程帮他解答。

输入描述:

第一行包含两个整数n和m，表示一个矩阵包含n行m列，用空格分隔。（ $1 \leq n \leq 10, 1 \leq m \leq 10$ ）从2到n+1行，每行输入m个整数（范围-231~231-1），用空格分隔，共输入n*m个数，表示第一个矩阵中的元素。

输出描述:

输出m行n列，为矩阵转置后的结果。每个数后面有一个空格。

示例1

输入

```
2 3
1 2 3
4 5 6
```

输出

```
1 4
2 5
3 6
```

参考代码:

```
#include <stdio.h>

int main()
{
    int n = 0;
    int m = 0;
    scanf("%d%d", &n, &m);
    int a[n][m];
    int i = 0;
    int j = 0;
    for(i=0; i<n; i++)
    {
        for(j=0; j<m; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
}
```



```

    }
    //转置
    for(i=0; i<m; i++){
        {
            for(j=0; j<n; j++){
                printf("%d ", a[j][i]);
            }
            printf("\n");
        }
    }
    return 0;
}

```

BC108-矩阵交换

题目描述

Kiki有一个矩阵，他想知道经过k次行变换或列变换后得到的矩阵。请编程帮他解答。

输入描述:

第一行包含两个整数n和m，表示一个矩阵包含n行m列，用空格分隔。（ $1 \leq n \leq 10, 1 \leq m \leq 10$ ）
从2到n+1行，每行输入m个整数（范围-231~231-1），用空格分隔，共输入n*m个数，表示第一个矩阵中的元素。
接下来一行输入k，表示要执行k次操作（ $1 \leq k \leq 5$ ）。接下来有k行，每行包括一个字符t和两个数a和b，中间用空格分隔，t代表需要执行的操作，当t为字符'r'时代表进行行变换，当t为字符'c'时代表进行列变换，a和b为需要互换的行或列（ $1 \leq a \leq b \leq n \leq 10, 1 \leq a \leq b \leq m \leq 10$ ）。

输出描述:

输出n行m列，为矩阵交换后的结果。每个数后面有一个空格。

示例1

输入

```

2 2
1 2
3 4
1
r 1 2

```

输出

```

3 4
1 2

```

参考代码:

```

#include <stdio.h>
int main()

```

```

{
    int m = 0; //列
    int n = 0; //行
    int arr[10][10] = { 0 };
    int cnt = 0; //操作的次数
    char op = 0; //代表操作的字符
    int a = 0;
    int b = 0;
    //读取数据
    scanf("%d%d", &n, &m);
    int i = 0;
    int j = 0;
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
        {
            scanf("%d", &arr[i][j]);
        }
    }
    scanf("%d", &cnt);
    for (i = 0; i < cnt; i++)
    {
        //读取操作指令
        //注释解释:
        //scanf("%d", &a);
        //scanf(" %c", &c); // 消耗 %d 后的所有后继空白符，然后读一个 char
        scanf(" %c %d %d", &op, &a, &b);
        //执行操作
        if (op == 'r')
        {
            //交换a和b行
            for (j = 0; j < m; j++)
            {
                int tmp = arr[a - 1][j];
                arr[a - 1][j] = arr[b - 1][j];
                arr[b - 1][j] = tmp;
            }
        }
        else if (op == 'c')
        {
            //交换a和b列
            for (j = 0; j < n; j++)
            {
                int tmp = arr[j][a - 1];
                arr[j][a - 1] = arr[j][b - 1];
                arr[j][b - 1] = tmp;
            }
        }
    }
    //输出
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < m; j++)
        {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }
}

```

```
return 0;    比特就业课，400小时就业课，保姆式就业服务
}
```

答案解析：

1. 如果是%c读取字符，一定要清理前面输入时留下的空白字符
2. 注意二维数组的下标

BC109-杨辉三角

题目描述

KiKi知道什么叫杨辉三角之后对杨辉三角产生了浓厚的兴趣，他想知道杨辉三角的前n行，请编程帮他解答。杨辉三角，本质上是二项式 $(a+b)^n$ 次方展开后各项的系数排成的三角形。其性质包括：每行的端点数为1，一个数也为1；每个数等于它左上方和上方的两数之和。

输入描述:

第一行包含一个整数数n。 ($1 \leq n \leq 30$)

输出描述:

包含n行，为杨辉三角的前n行，每个数输出域宽为5。

示例1

输入

6

输出

```
1
1  1
1  2  1
1  3  3  1
1  4  6  4  1
1  5 10 10 5  1
```

参考代码：

```
#include <stdio.h>

int main()
{
    int arr[30][30] = {0};
    int i = 0;
    int j = 0;
    int n = 0;
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
```

```
for(j=0; j<=i; j++)
{
    if(i == j)
        arr[i][j] = 1;
    if(j==0)
        arr[i][j] = 1;
    if(i>1 && j>0)
        arr[i][j] = arr[i-1][j-1]+arr[i-1][j];
}
}
for(i=0; i<n; i++)
{
    for(j=0; j<=i; j++)
    {
        printf("%5d", arr[i][j]);
    }
    printf("\n");
}
return 0;
}
```

BC110-井字棋

题目描述

KiKi和BoBo玩“井”字棋。也就是在九宫格中，只要任意行、列、或者任意对角线上面出现三个连续相同的棋子，就能获胜。请根据棋盘状态，判断当前输赢。

输入描述:

三行三列的字符元素，代表棋盘状态，字符元素用空格分开，代表当前棋盘，其中元素为K代表KiKi玩家的棋子，为O表示没有棋子，为B代表BoBo玩家的棋子。

输出描述:

如果KiKi获胜，输出“KiKi wins!”;
如果BoBo获胜，输出“BoBo wins!”;
如果没有获胜，输出“No winner!”。

示例1

输入

```
K O B
O K B
B O K
```

输出

```
KiKi wins!
```

参考代码:

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

```
int main()
{
    char arr[3][3] = {0};
    int i = 0;
    int j = 0;
    char flag = 'c';
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            scanf("%c", &arr[i][j]);
            getchar();
        }
    }
    for(i=0; i<3; i++)
    {
        if(arr[i][0]==arr[i][1] &&arr[i][1]==arr[i][2])
        {
            flag = arr[i][1];
            break;
        }
        if(arr[0][i]==arr[1][i] &&arr[1][i]==arr[2][i])
        {
            flag = arr[1][i];
            break;
        }
    }

    if(arr[0][0]==arr[1][1] &&arr[1][1]==arr[2][2])
    {
        flag = arr[1][1];
    }
    if(arr[0][2]==arr[1][1] &&arr[1][1]==arr[2][0])
    {
        flag = arr[1][1];
    }

    if(flag == 'K')
        printf("KiKi wins!");
    else if(flag == 'B')
        printf("BoBo wins!");
    else
        printf("No winner!");

    return 0;
}
```

联系鹏哥：15596668862 (同微信) 比特就业课，400小时就业课，保姆式就业服务

[完整的C语言教程链接](#)

比特就业课