2D Array Diagonals in C

Write a C Program to read the values for a **square** 2D array and print its main/principal and secondary diagonals.

Input Format

- First line specifies the **no: of rows** (m) and and the **no: of columns** (n) in the 2D Array.
- Second line contains m*n integers that signify the **values** for the array.

Constraints

Make sure that the given 2D array is a **square matrix**. If it is not, then print "*Error*"

Output Format

- If it is not a square matrix, then print "*ERROR*"

Else,

- First line displays **main** diagonal elements.
- Second line displays **secondary** diagonal elements.

Sample Input 0

```
3 3
1 6 0 2 7 3 3 4 9
```

Sample Output 0

```
1 7 9
0 7 3
```

Explanation 0

The given matrix is,

```
1 6 0
2 7 3
3 4 9
```

So, the main diagonal elements are : 1 7 9
and the secondary diagonal elements are : 0 7 3

Sample Input 1

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

Sample Output 1

```
ERROR
```

Explanation 1

The given matrix is not a square matrix.

Current Buffer (saved locally, editable)

C

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main()
{
    int m,n,i,j,a[20][20];
    scanf("%d%d",&m,&n);
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    if(m==n)
    {
            for(i=0;i<m;i++)
            {
                for(j=0;j<n;j++)
                {
                    if (i==j)
                        printf("%d ",a[i][j]);
                }
            }
            printf("\n");
            for(i=0;i<m;i++)
            {
                for(j=0;j<n;j++)
                {
                    if((i+j) == (n-1))
                        printf("%d ",a[i][j]);
                }
            }
    }
    else
        printf("ERROR");

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}

```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔ Testcase 1 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
3 3
1 6 0 2 7 3 3 4 9
```

Your Output (stdout)

```
1 7 9
0 7 3
```

Expected Output

```
1 7 9
0 7 3
```