# Arrays 2D - Transpose Matrix

Sheldon Cooper has a square-shaped puzzle made up of small square pieces containing numbers on them. He wants to rearrange the puzzle by changing the elements of a row into a column element and the column element into a row element without altering the shape of the puzzle. Help Sheldon solves this puzzle. Write a program to find the transpose of the given 2D matrix.

### Input Format

The first line consists of an integer which represents the number of rows and columns. The next line consists of the elements in the matrix.

### Constraints

NA

### Output Format

Output prints the transpose of the input matrix.

### Sample Input 0

```
3
1 2 3
4 5 6
7 8 9
```

### Sample Output 0

```
1 2 3
4 5 6
7 8 9
Transpose matrix is:
1 4 7
2 5 8
3 6 9
```

### Sample Input 1

```
4
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

### Sample Output 1

```
1 2 3 4
5 6 7 8
```

```
9 10 11 12
13 14 15 16
Transpose matrix is:
1 5 9 13
2 6 10 14
3 7 11 15
4 8 12 16
```

**Submissions:** 574
**Max Score:** 100
**Difficulty:** Medium

Rate This Challenge:
☆☆☆☆☆

More

C

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

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

    for(i=0;i<n;i++)
    {
        for (j=0;j<n;j++)
        {
            printf("%d ",arr[i][j]);
        }
         printf("\n");
    }

    printf("Transpose matrix is:\n");
     for(i=0;i<n;i++)
    {
        for (j=0;j<n;j++)
        {
            printf("%d ",arr[j][i]);
        }
         printf("\n");
    }
    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
1 2 3
4 5 6
7 8 9
```

Your Output (stdout)

```
1 2 3
4 5 6
7 8 9
Transpose matrix is:
1 4 7
2 5 8
3 6 9
```

Expected Output

```
1 2 3
4 5 6
7 8 9
Transpose matrix is:
1 4 7
2 5 8
3 6 9{-truncated-}
```