

회전 행렬

Description

이미지를 90도 오른쪽으로 회전시키는 것은 행렬을 90도 오른쪽으로 회전하는 것과 같다.

주어진 N \* N 정방행렬을 90도 오른쪽으로 회전시켜보자.

예를 들어, 3 \* 3 행렬이 다음과 같이 주어졌다고 해보자.

1 2 3  
4 5 6  
7 8 9

이 행렬을 90도 오른쪽으로 회전시키면 다음과 같다.

7 4 1  
8 5 2  
9 6 3

N \* N 정방행렬이 주어졌을때, 이 행렬을 90도 오른쪽으로 회전해서 출력하시오.

다음 코드에서 rotate() 함수를 구현하여 제출하시오.

```
#include <stdio.h>
#define MAX 100

/* rotates the matix A into the matrix B. */
void rotate(int n, int A[][MAX], int B[][MAX]);

int main()
{
    int n, A[MAX][MAX], B[MAX][MAX];
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            scanf("%d", &A[i][j]);
    rotate(n, A, B);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++)
            printf("%d ", B[i][j]);
        printf("\n");
    }
}
```

Input

첫 번째 줄에 양의 정수 N이 주어진다. (1 <= N <= 100)

두 번째 줄부터 N번째 줄에 걸쳐, N개의 열을 가진 행이 출력된다.

행렬의 원소값은 1보다 크거나 같고, 100보다 작거나 같다.

Output

첫 번째 줄부터 N번째 줄까지 주어진 행렬을 90도 회전한 행렬을 출력한다.

Sample Input 1

3  
1 2 3  
4 5 6  
7 8 9

Sample Output 1

7 4 1  
8 5 2  
9 6 3

Sample Input 2

10  
81 48 24 80 34 65 46 86 37 14  
14 55 14 47 65 75 95 42 26 83  
25 38 70 95 65 88 53 10 11 15  
37 25 76 66 24 22 23 79 75 36  
67 97 45 39 52 48 59 44 71 63  
28 37 83 88 35 92 95 37 13 49  
64 41 65 93 59 52 79 37 85 28  
17 23 52 41 30 14 58 42 43 22  
91 66 86 19 93 69 10 68 35 89  
33 41 25 65 94 65 43 98 40 83

Sample Output 2

33 91 17 64 28 67 37 25 14 81  
41 66 23 41 37 97 25 38 55 48  
25 86 52 65 83 45 76 70 14 24  
65 19 41 93 88 39 66 95 47 80  
94 93 30 59 35 52 24 65 65 34  
65 69 14 52 92 48 22 88 75 65  
43 10 58 79 95 59 23 53 95 46  
98 68 42 37 37 44 79 10 42 86  
40 35 43 85 13 71 75 11 26 37  
83 89 22 28 49 63 36 15 83 14

```
1 void rotate(int n, int A[][MAX], int B[][MAX])
2 {
3     for (int x = 0; x < n; x++)
4     {
5         for (int y = 0; y < n; y++)
6         {
7             // if (n - x - 1 < 0)
8             // {
9             //     x = n - 1;
10            //     B[y][n - x - 1] = A[x][y];
11            // }
12            // else
13            // {
14            //     B[y][n - x - 1] = A[x][y];
15            // }
16            B[y][n - x - 1] = A[x][y];
17        }
18    }
19 }
```