Pointer & dynamic array ----- Rotate matrix

Description

You are given an n x n 2D matrix representing an image, rotate the image by 90 degrees (clockwise).

Input (use cin)

The first line of the input file specifies the size n of the matrix. Starting from the second line, the input consists of a 2D square matrix.

Constraints

1 ≤ n≤ 20

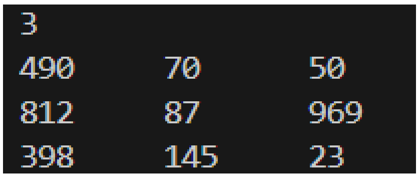
0 ≤ matrix element ≤ 999

Output (use cout)

Output the result of the rotated matrix.

Example

Input:

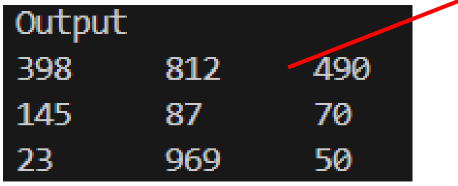


```
3
490     70      50
812     87      969
398     145     23
```

Output

Numbers are separated by "\t"

"\t"



```
Output
398     812     490
145     87      70
23      969     50
```

Notice: Please follow the rules below, or you will get 0 point.

- □ The objection of this exercise is to **practice dynamic array**
    - □ You should use **new** operator and **delete**
    - □ Check that you are successfully freeing the memory by **Valgrind**
    - □ TA will check your code

Compile & Execute:

Compile: g++ Mid01.cpp -o Mid01

Execute: ./Mid01

OJ:/home/share/demo_OOP112_2 Mid 01

Valgrind
(Please ensure that both Mid01 and input.txt are in the same folder.)

Case1:

```
valgrind --leak-check=full -s --show-leak-kinds=all --track-origins=yes ./Mid01 < input1.txt
```

Case2:

```
valgrind --leak-check=full -s --show-leak-kinds=all --track-origins=yes ./Mid01 < input2.txt
```

You must pass the case mentioned above.