

홀수 마방진

Description

N * N 크기의마방진(magic square)은, 가로, 세로, 대각선 방향의 수를 모두 더하면 모두 같은 값이 나오는 2차원 행렬을 말한다.

단, 마방진 행렬의 모든 수는 1부터 N^2까지의 수가 딱 한 번씩만 나타난다.

일반적으로 N이 2일 때를 제외하고는 항상 마방진이 존재한다고 알려져 있으며, N이 홀수인 마방진을 구하는 간단한 방법이 알려져 있다.

아래 사이트를 참조하여 루베르가 만든 홀수 마방진을 구하는 방법을 따라 마방진을 만들어서 출력하시오.

https://ko.wikipedia.org/wiki/%EB%A7%88%EB%B0%A9%EC%A7%84#%EB%A7%88%EB%B0%A9%EC%A7%84_%EB%A7%8C%EB%93%A4%EA%B8%B0

아래의 코드에서 magic() 함수를 완성하여 제출하시오.

```
#include <stdio.h>

#define MAX 100

int square[MAX][MAX];

/* Make n * n magic square in a 2-dimensional array, square[].
 * It is guaranteed that n is always odd.
 */
void magic(int n);

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

Input

첫 번째 줄에 양의 정수 N이 주어진다.

단, N은 2보다 크고 100보다 작은 홀수이다.

Output

N * N 크기의 마방진을 출력한다.

한 줄에 N개의 열을 가진 행을 하나씩 출력한다.

Sample Input 1

3

Sample Output 1

8 1 6
3 5 7
4 9 2

Sample Input 2

5

Sample Output 2

17 24 1 8 15
23 5 7 14 16
4 6 13 20 22
10 12 19 21 3
11 18 25 2 9

```
1 void magic(int n)
2 {
3     int row = 0, col = n / 2;
4     int num = 1;
5     int maxNum = n * n;
6
7     for (; num <= maxNum; num++)
8     {
9         square[row][col] = num;
10
11         int nextRow = (row - 1 + n) % n;
12         int nextCol = (col + 1) % n;
13
14         if (square[nextRow][nextCol] != 0)
15         {
16             row = (row + 1) % n;
17         }
18         else
19         {
20             row = nextRow;
21             col = nextCol;
22         }
23     }
24 }
25
```