

On an $N \times N$ board, the numbers from 1 to $N*N$ are written *boustrophedonically* starting from the bottom left of the board, and alternating direction each row. For example, for a 6 x 6 board, the numbers are written as follows:

```
int snakesAndLadders(vector<vector<int>> &board)
{
    n = board.size();
    m = board[0].size();

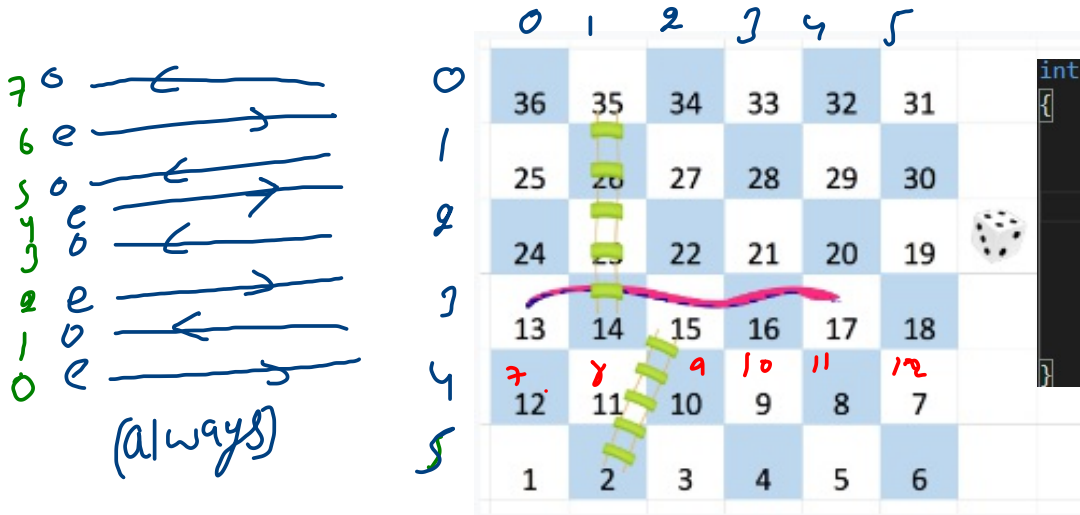
    queue<int> que;
    que.push(1);

    vector<int> vis(n * m + 1, false);
    int moves = 0;

    while (que.size() != 0)
    {
        int size = que.size();
        while (size-- > 0)
        {
            int step = que.front();
            que.pop();
            if (vis[step])
                continue; //cycle.

            if (step == n * m)
                return moves;

            vis[step] = true; //mark it true.
            for (int i = 1; i <= 6 && step + i <= n * m; i++)
            {
                int num = step + i;
                int value = getValueOverBoard(board, num);
                if (value > 0)
                    num = value;
                if (vis[step])
                    que.push(num);
            }
        }
        moves++;
    }
    return -1;
}
```

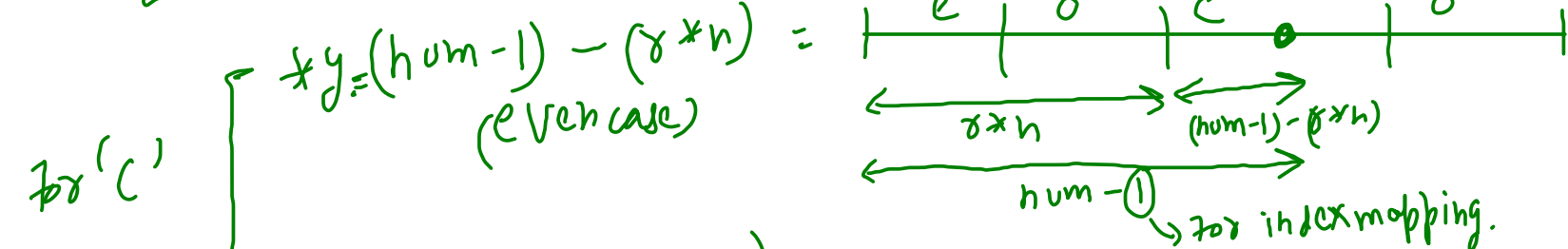
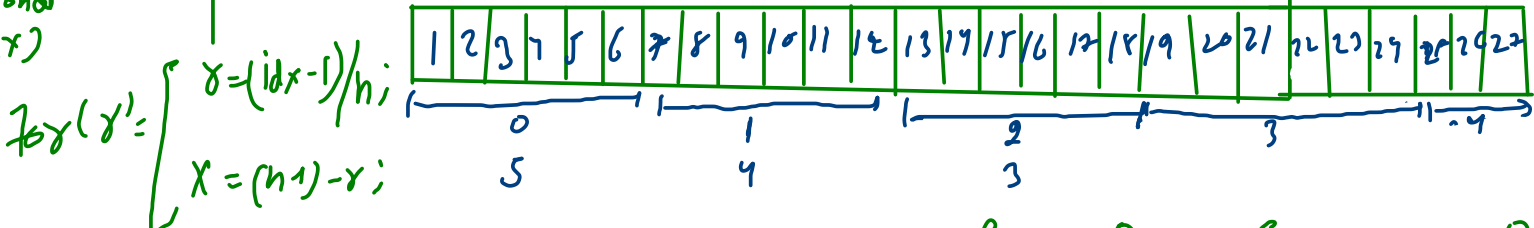


```
int getValueOverBoard(vector<vector<int>> &board, int num)
{
    int r = (num - 1) / n;

    int x = (n - 1) - r;
    int y = r % 2 == 0 ? num - 1 - r * n : n - (num - r * n);

    board[x][y] = num;
    return board[x][y];
}
```

(according to boustrophedon matrix)



for 'c'

$$x = (num - 1) / n;$$
$$y = (num - 1) - (x * n);$$
$$y = (n - (num - x * n));$$

(even case)

(odd case)

