Spiral traversal

Description

Given a matrix of size n by n. Traverse and print the matrix in spiral form.

Input

Input Format

First-line contains n

The next n lines contain the matrix

Constraints

n <= 1000

Ai <= 10000

Output

Print the matrix in a single line traversing it spirally

Sample Input 1

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

Sample Output 1

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

```
import java.util.*;
import java.io.*;

public class Main{
    static class FastReader{
        BufferedReader br;
        StringTokenizer st;
```

```
        public FastReader(){
            br=new BufferedReader(new InputStreamReader(System.in));
        }
        String next(){
            while(st==null || !st.hasMoreTokens()){
                try {
                    st=new StringTokenizer(br.readLine());
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            return st.nextToken();
        }
        int nextInt(){
            return Integer.parseInt(next());
        }
        long nextLong(){
            return Long.parseLong(next());
        }
        double nextDouble(){
            return Double.parseDouble(next());
        }
        String nextLine(){
            String str="";
            try {
                str=br.readLine().trim();
            } catch (Exception e) {
                e.printStackTrace();
            }
            return str;
        }
    }
    static class FastWriter {
    private final BufferedWriter bw;

    public FastWriter() {
      this.bw = new BufferedWriter(new OutputStreamWriter(System.out));
    }

    public void print(Object object) throws IOException {
      bw.append("" + object);
    }

    public void println(Object object) throws IOException {
      print(object);
      bw.append("\n");
    }

    public void close() throws IOException {
      bw.close();
    }
  }
    public static void main(String[] args) {
        try {
            FastReader sc=new FastReader();
            FastWriter out = new FastWriter();
            int i =sc.nextInt();
            int[][] mat = new int[i][i];
```

```
            for(int x=0;x<i;x++){
                for(int y=0;y<i;y++){
                    mat[x][y]=sc.nextInt();
                }
            }
            int left = 0;
        int right = i-1;
        int top = 0;
        int botom = i-1;
        int count=0;
        ArrayList<Integer> arr = new ArrayList<>();

        while(count < i*i){
            for(int x = left; x<=right; x++){
                arr.add(mat[top][x]);
                count++;
            }
            top++;
            for(int x=top;x<=botom;x++){
                arr.add(mat[x][right]);
                count++;
            }
            right--;
            for(int x=right;x>=left;x--){
                arr.add(mat[botom][x]);
                count++;
            }
            botom--;
            for(int x=botom;x>=top;x--){
                arr.add(mat[x][left]);
                count++;
            }
            left++;
        }
        for(int x: arr){
            System.out.print(x+" ");
        }
            out.close();
        } catch (Exception e) {
            e.printStackTrace() ;
            return;
        }
    }
}
```