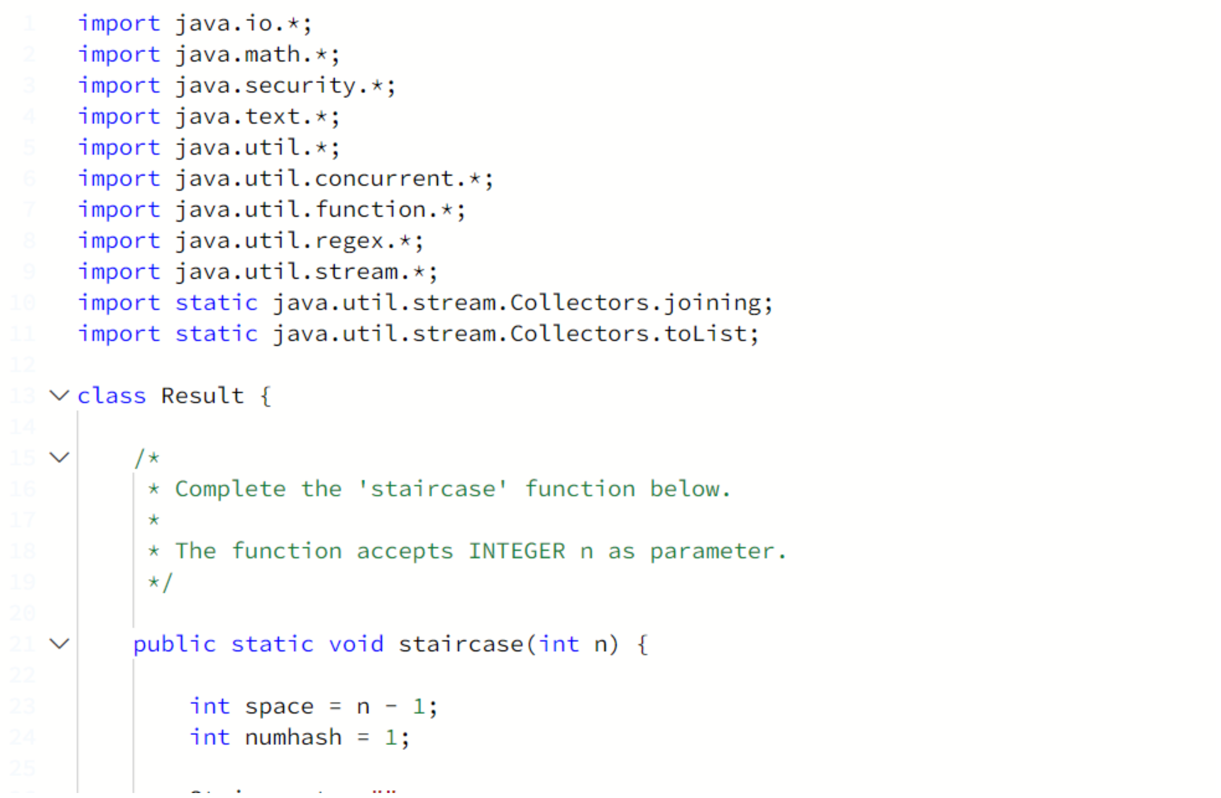
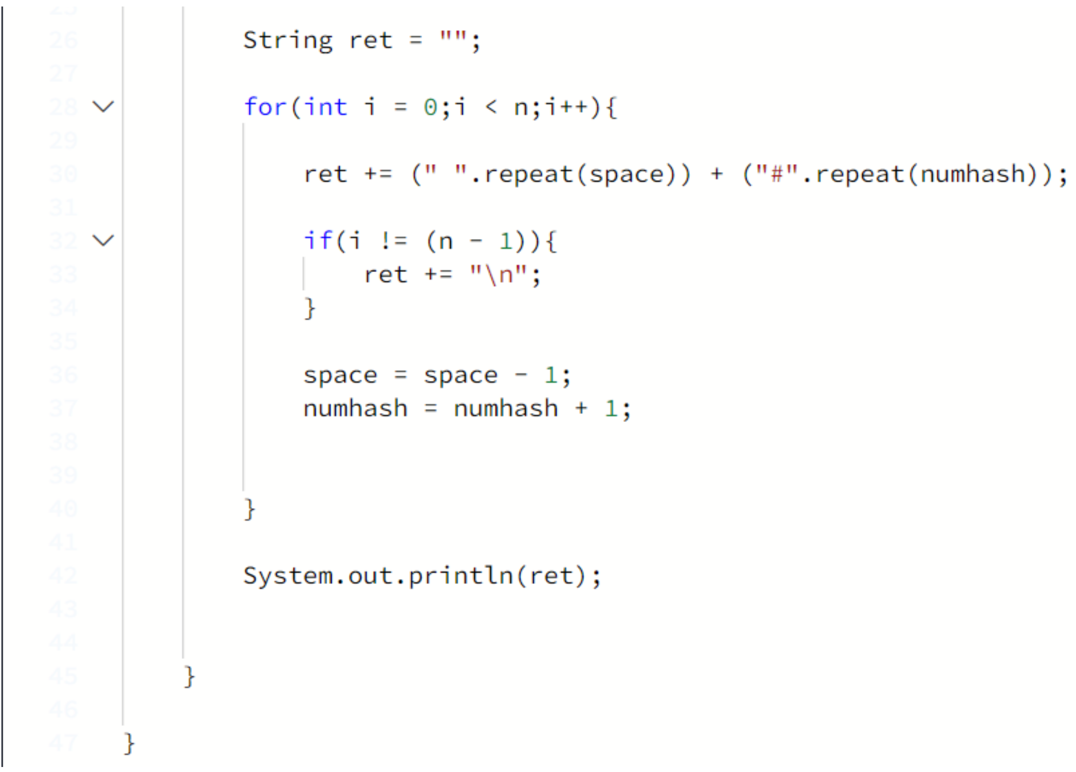
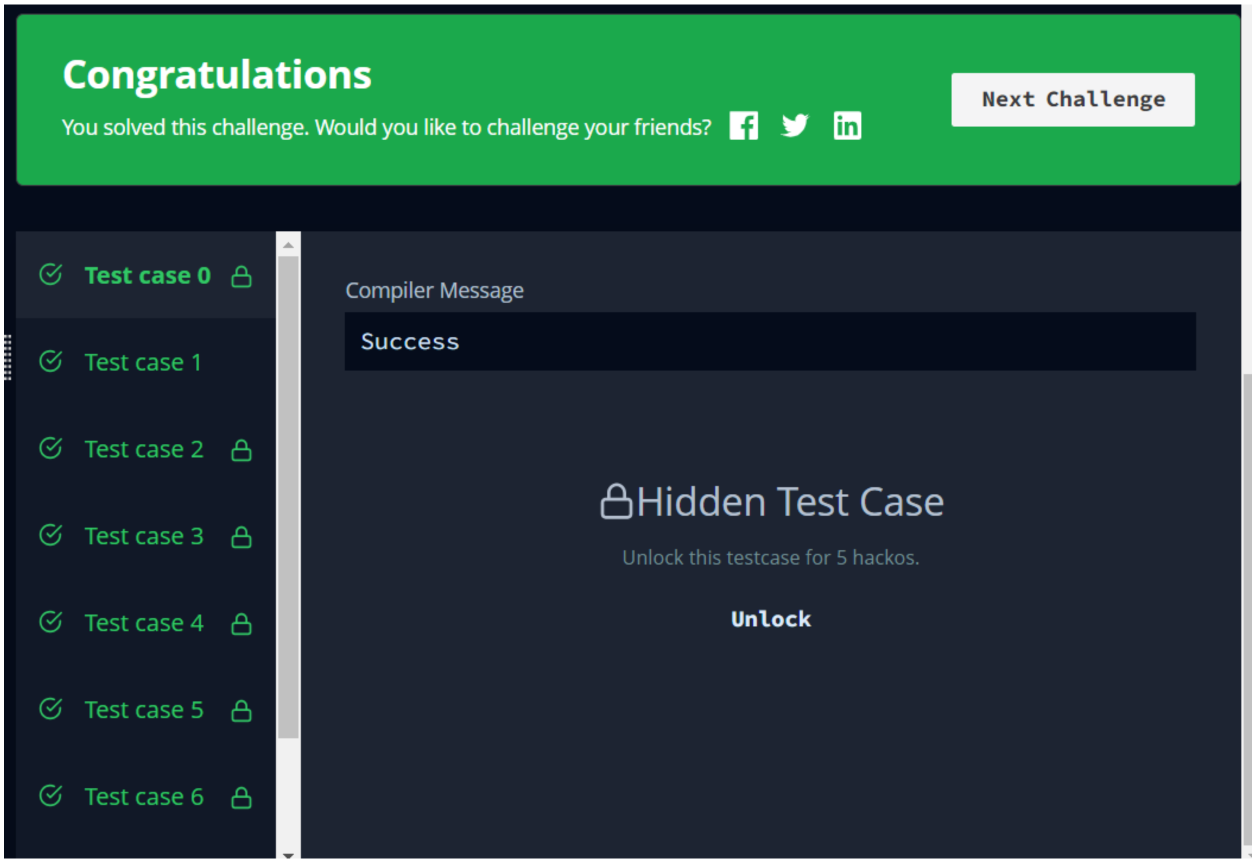
STAIRCASE PROBLEM



```java
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'staircase' function below.
     *
     * The function accepts INTEGER n as parameter.
     */

    public static void staircase(int n) {

        int space = n - 1;
        int numhash = 1;

```



```java
        String ret = "";

        for(int i = 0;i < n;i++){

            ret += (" ".repeat(space)) + ("#".repeat(numhash));

            if(i != (n - 1)){
                ret += "\n";
            }

            space = space - 1;
            numhash = numhash + 1;


        }

        System.out.println(ret);


    }

}
```


Congratulations

You solved this challenge. Would you like to challenge your friends?

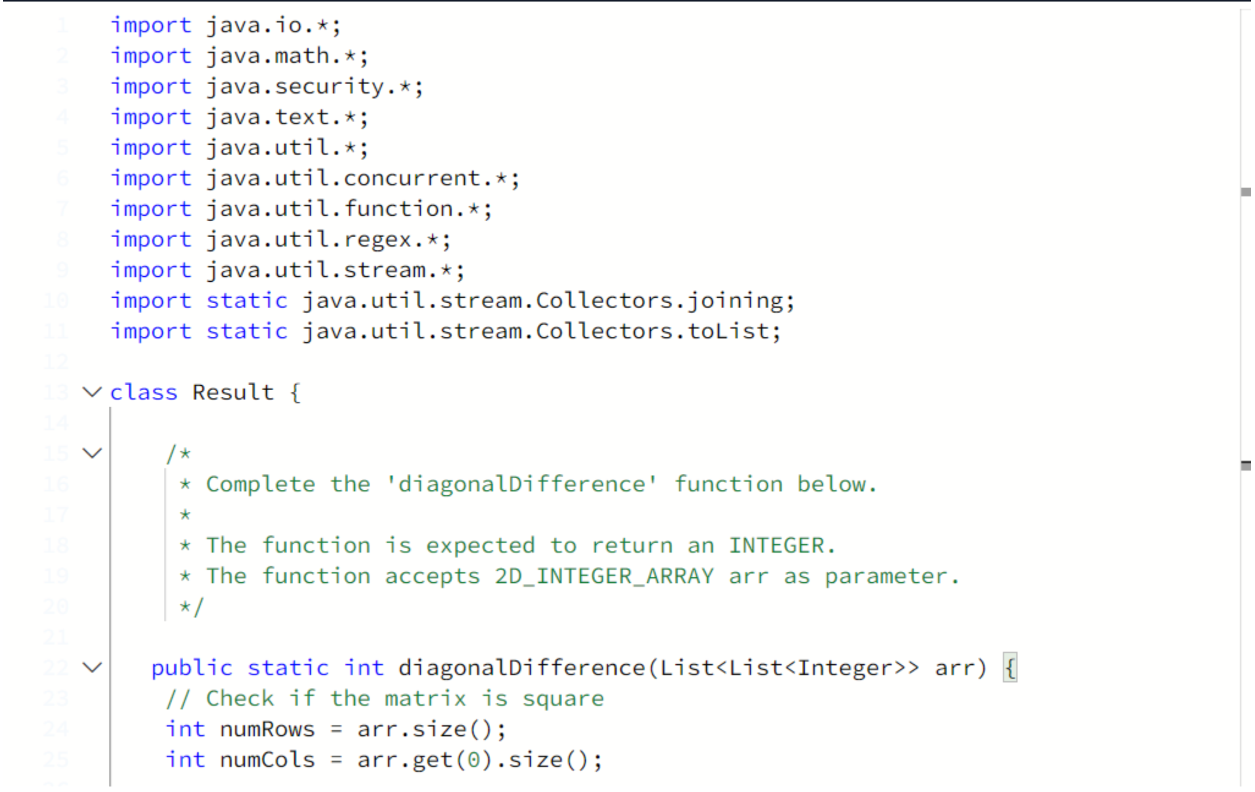
Compiler Message

```
Success
```

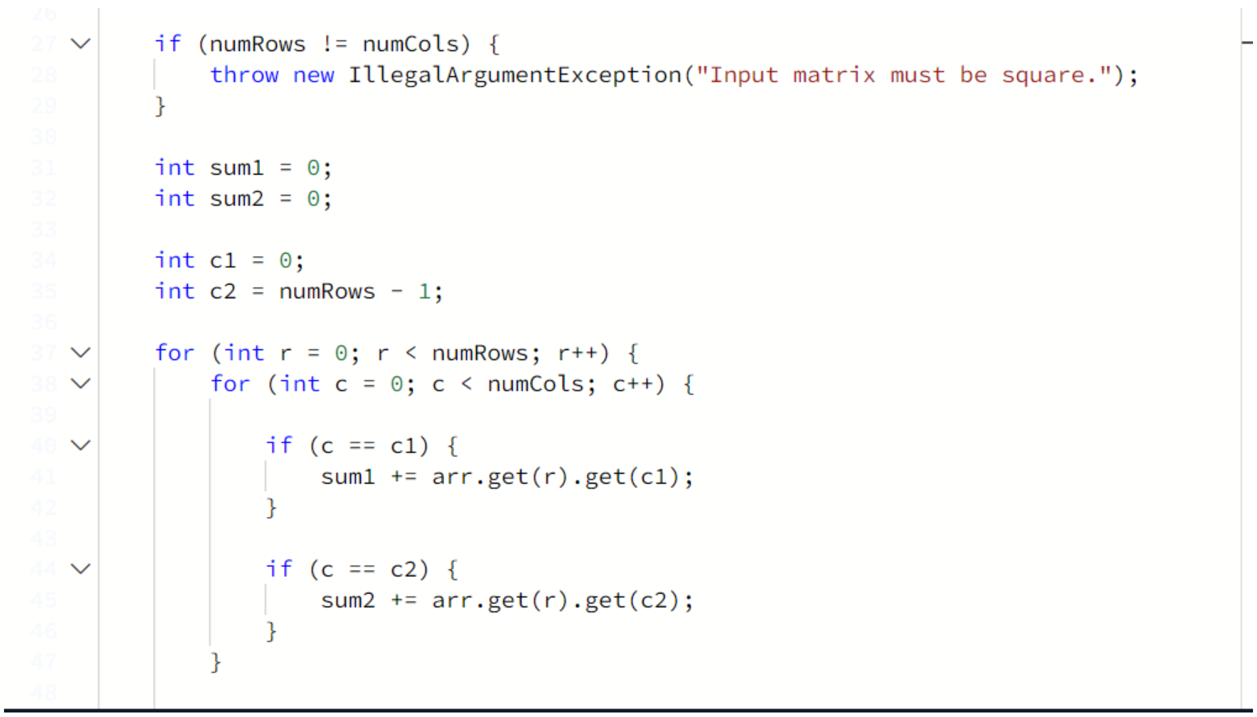
Hidden Test Case

Unlock this testcase for 5 hackos.

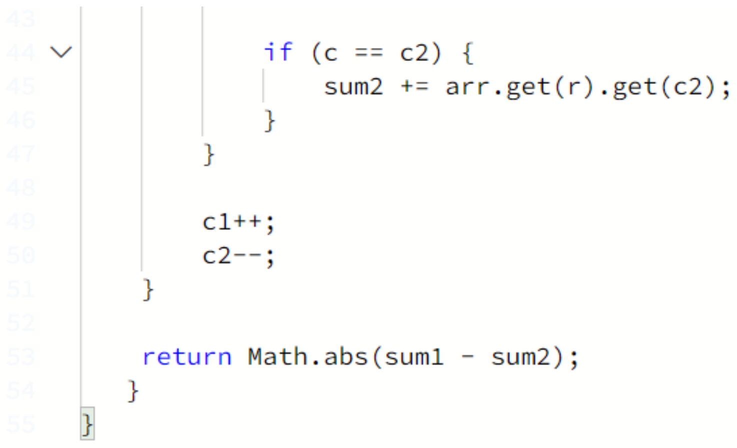
DIAGONAL DIFFERENCE



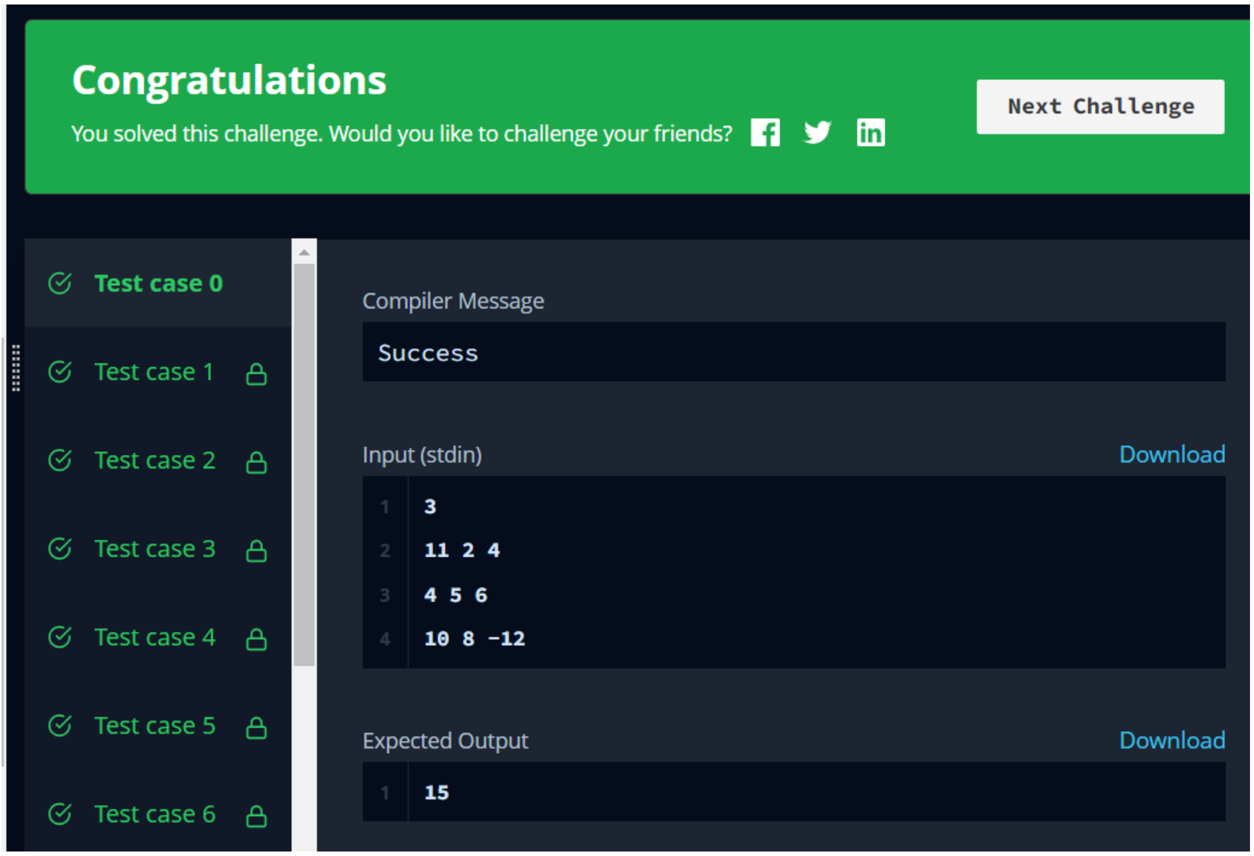
```java
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'diagonalDifference' function below.
     *
     * The function is expected to return an INTEGER.
     * The function accepts 2D_INTEGER_ARRAY arr as parameter.
     */

   public static int diagonalDifference(List<List<Integer>> arr) {
    // Check if the matrix is square
    int numRows = arr.size();
    int numCols = arr.get(0).size();
```



```java

    if (numRows != numCols) {
        throw new IllegalArgumentException("Input matrix must be square.");
    }

    int sum1 = 0;
    int sum2 = 0;

    int c1 = 0;
    int c2 = numRows - 1;

    for (int r = 0; r < numRows; r++) {
        for (int c = 0; c < numCols; c++) {

            if (c == c1) {
                sum1 += arr.get(r).get(c1);
            }

            if (c == c2) {
                sum2 += arr.get(r).get(c2);
            }
        }

```



```
                if (c == c2) {
                    sum2 += arr.get(r).get(c2);
                }
            }

            c1++;
            c2--;
        }

        return Math.abs(sum1 - sum2);
    }
}
```



Congratulations

You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

Input (stdin)

```
3
11 2 4
4 5 6
10 8 -12
```

Expected Output

```
15
```