2D Array - DS

You have successfully solved 2D Array - DS Share Tweet

Try the next challenge

Problem | Submissions | Leaderboard | Editorial

RATE THIS CHALLENGE

Given a 6×6 2D Array, arr:

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

We define an hourglass in A to be a subset of values with indices falling in this pattern in arr's graphical representation:

```
a b c
  d
e f g
```

There are 16 hourglasses in arr, and an hourglass sum is the sum of an hourglass' values. Calculate the hourglass sum for every hourglass in arr, then print the maximum hourglass sum.

For example, given the 2D array:

```
-9 -9 -9  1 1 1
 0 -9  0  4 3 2
-9 -9 -9  1 2 3
 0  0  8  6 6 0
 0  0  0 -2 0 0
 0  0  1  2 4 0
```

We calculate the following 16 hourglass values:

```
-63, -34, -9, 12,
-10, 0, 28, 23,
-27, -11, -2, 10,
9, 17, 25, 18
```

Our highest hourglass value is 28 from the hourglass:

```
0 4 3
  1
8 6 6
```

Note: If you have already solved the Java domain's Java 2D Array challenge, you may wish to skip this challenge.

Function Description

Complete the function hourglassSum in the editor below. It should return an integer, the maximum hourglass sum in the array.

hourglassSum has the following parameter(s):

- arr: an array of integers

Input Format

Each of the 6 lines of inputs $arr[i]$ contains 6 space-separated integers $arr[i][j]$.

Constraints

- $-9 \leq arr[i][j] \leq 9$
- $0 \leq i, j \leq 5$

Output Format

Print the largest (maximum) hourglass sum found in arr.

Sample Input

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 2 4 4 0
0 0 0 2 0 0
0 0 1 2 4 0
```

Sample Output

```
19
```

Explanation

arr contains the following hourglasses:

```
1 1 1   1 1 0   1 0 0   0 0 0
  1       0       0       0
1 1 1   1 1 0   1 0 0   0 0 0

0 1 0   1 0 0   0 0 0   0 0 0
  1       1       0       0
0 0 2   0 2 4   2 4 4   4 4 0

1 1 1   1 1 0   1 0 0   0 0 0
  0       2       4       4
0 0 0   0 0 2   0 2 0   2 0 0

0 0 2   0 2 4   2 4 4   4 4 0
  0       0       2       0
0 0 1   0 1 2   1 2 4   2 4 0
```

The hourglass with the maximum sum (19) is:

```
2 4 4
  2
1 2 4
```

Current Buffer (saved locally, editable)

Java 8

```java
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

public class Solution {

    // Complete the hourglassSum function below.
    static int hourglassSum(int[][] arr) {
        int greatestSum = -63;
        for(int i=1; i<5; i++) {
            for(int j=1; j<5; j++) {
                int sum = arr[i-1][j-1] + arr[i-1][j] + arr[i-1][j+1]
                                    + arr[i][j]
                    + arr[i+1][j-1] + arr[i+1][j] + arr[i+1][j+1];

                if (sum > greatestSum) {
                    greatestSum = sum;
                }
            }
        }
        return greatestSum;
    }

    private static final Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) throws IOException {
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));

        int[][] arr = new int[6][6];

        for (int i = 0; i < 6; i++) {
            String[] arrRowItems = scanner.nextLine().split(" ");
            scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

            for (int j = 0; j < 6; j++) {
                int arrItem = Integer.parseInt(arrRowItems[j]);
                arr[i][j] = arrItem;
            }
        }

```

```
45          int result = hourglassSum(arr);
46
47          bufferedWriter.write(String.valueOf(result));
48          bufferedWriter.newLine();
49
50          bufferedWriter.close();
51
52          scanner.close();
53      }
54  }
55
```

Line: 55 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations

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

Testcase 0 | Testcase 1 | Testcase 2 | Testcase 3 | Testcase 4 | Testcase 5 | Testcase 6

Input (stdin) Download

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
```

Expected Output Download

```
19
```

Compiler Message

```
Success
```