2D Array - DS ★

Problem Submissions Leaderboard Editorial 🖰

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 0 0

An hourglass in $m{A}$ is a subset of values with indices falling in this pattern in $m{arr}$'s graphical representation:

a b c d

e f g

There are 16 hourglasses in arr. 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. The array will always be 6×6 .

Example

arr =

-9 -9 -9 1 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 0 1 2 4 0

The **16** hourglass sums are:

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

The highest hourglass sum is $\mathbf{28}$ from the hourglass beginning at row $\mathbf{1}$, column $\mathbf{2}$:

0 4 3

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.

hourglassSum has the following parameter(s):

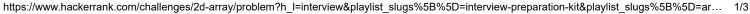
• int arr[6][6]: an array of integers

Returns

• int: the maximum hourglass sum

Input Format

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



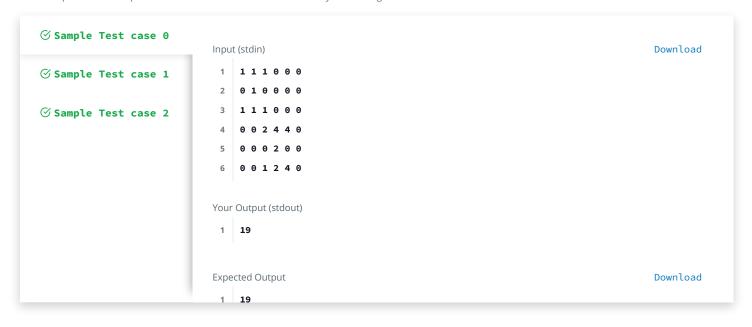
```
Constraints
• -9 \leq arr[i][j] \leq 9
• 0 \le i, j \le 5
Output Format
Print the largest (maximum) hourglass sum found in arr.
Sample Input
  1 1 1 0 0 0
  0 1 0 0 0 0
  1 1 1 0 0 0
  0 0 2 4 4 0
  0 0 0 2 0 0
  0 0 1 2 4 0
Sample Output
  19
Explanation
arr contains the following hourglasses:
    111 110 100 000
    0101000000000
    0 0 2 0 2 4 2 4 4 4 4 0
    111 110 100 000
    000002020200
    \begin{smallmatrix} 0 & 0 & 2 & & 0 & 2 & 4 & & 2 & 4 & 4 & & 4 & 4 & 0 \end{smallmatrix}
    The hourglass with the maximum sum (19) is:
  2 4 4
    2
  1 2 4
```

```
Change Theme
                                                                                  JavaScript (Node.js)
22
23
    function readLine() {
         return inputString[currentLine++];
24
25
26
27
    // Complete the hourglassSum function below.
28
    function hourglassSum(arr) {
29
30
             let m = arr
31
             let hour = []
```

```
Tor (let 1 = 1; 1 < m.lengtn-1; 1++){</pre>
        33
                                                                                 for(let j = 1; j <m[0].length-1; j++){</pre>
       34
       35
                                                                                                        \text{var suma = } m[i-1][j-1] \ + \ m[i-1][j] \ + \ m[i-1][j+1] \ + \ m[i][j] \ + \ m[i+1][j-1] \ + \ 
                                   [i+1][j] + m[i+1][j+1]
      36
                                                                                                       hour.push(suma)
       37
                                                                                  }
        38
                                                           let maximo = Math.max(...hour)
        39
       40
                                                           return maximo;
                                  }
       41
       42
        43
                                    function main() {
                                                           const ws = fs.createWriteStream(process.env.OUTPUT_PATH);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  Line: 30 Col: 20
☐ Test against custom input
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Run Code
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    Submit Code
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.



Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature

