

Day 11: 2D-Arrays + More Review!

Problem Statement

Welcome to Day 11! Review everything we've learned so far by making a library catalogue [in this video](#), or just jump right into the problem. We haven't discussed *2D Arrays* in this series, but they're very similar to the regular *1D Arrays* you're likely familiar with. If you are working in Java, check out [Oracle's documentation](#). Similar documentation for *2D Arrays* in other popular languages is easily found on the internet.

Given a 6×6 *2D Array*, A :

```
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 can find 16 hourglasses in A ; we define an hourglass in A to be a subset of values with indexes falling in this pattern in A 's graphical representation:

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

The *sum of an hourglass* is the sum of the values within it.

Your task is to calculate the sum of every hourglass in some *2D Array*, A , and print the *largest value* (maximum of the sums) as your answer.

Input Format

There are 6 lines of input, where each line contains 6 space-separated integers describing *2D Array* A ; every value in A will be in the inclusive range of -9 to 9 .

Constraints

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

Output Format

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

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

Sample Input \$A\$ 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 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
```