

## Objective

Today, we're building on our knowledge of *Arrays* by adding another dimension. Check out the [Tutorial](#) tab for learning materials and an instructional video!

## Context

Given a  $6 \times 6$  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 define an hourglass in  $A$  to be a subset of values with indices falling in this pattern in  $A$ 's graphical representation:

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

There are **16** hourglasses in  $A$ , and an *hourglass sum* is the sum of an hourglass' values.

## Task

Calculate the hourglass sum for every hourglass in  $A$ , then print the *maximum* hourglass sum.

## 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

$A$  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   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
```