

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
```

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

```
abc
 d
efg
```

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
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
```

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 **28** from the hourglass beginning at row **1**, column **2**:

```
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.

`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

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

Output Format

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

Sample Input

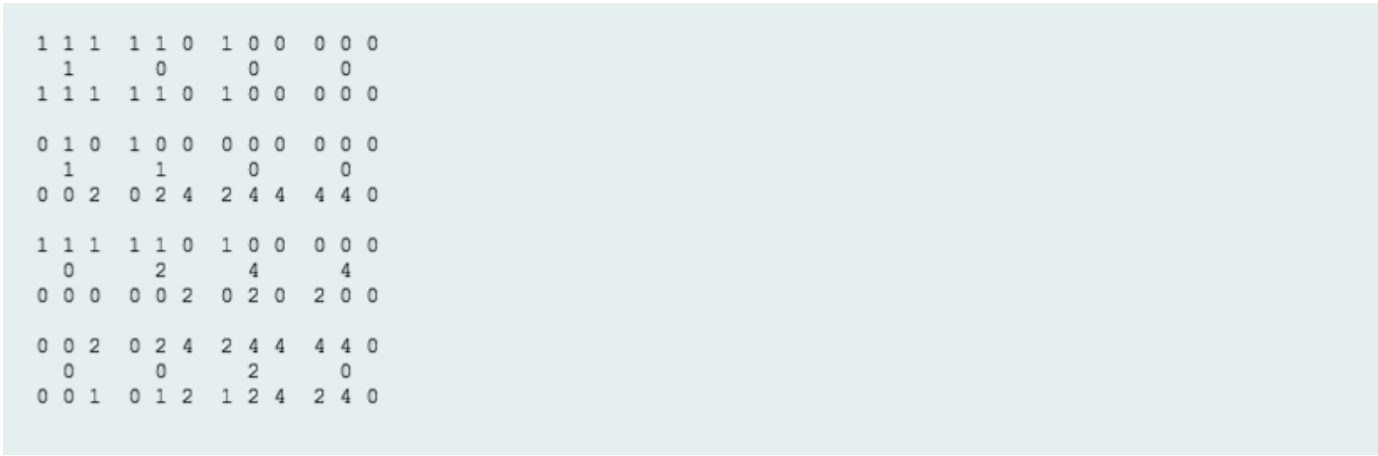
```
1 1 1 0 0
0 1 0 0 0
1 1 1 0 0
0 0 2 4 4
0 0 0 2 0
0 0 1 2 4
```

Sample Output

19

Explanation

arr contains the following hourglasses:



The hourglass with the maximum sum (**19**) is:

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