

Problem:

Context

Given a *2D Array*, :

```
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 to be a subset of values with indices falling in this pattern in 's graphical representation:

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

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

Task

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

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

Input Format

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

Constraints

-
-

Output Format

Print the largest (maximum) hourglass sum found in .

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

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 () is:

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

Solution:

```
#include <map>
#include <set>
#include <list>
#include <cmath>
#include <ctime>
#include <deque>
#include <queue>
#include <stack>
#include <string>
#include <bitset>
#include <cstdio>
#include <limits>
#include <vector>
#include <climits>
#include <cstring>
#include <cstdlib>
#include <fstream>
```

```

#include <numeric>
#include <sstream>
#include <iostream>
#include <algorithm>
#include <unordered_map>

using namespace std;

int main(){
    vector< vector<int> > array(6,vector<int>(6));
    for(int arr_i = 0;arr_i < 6;arr_i++){
        for(int arr_j = 0;arr_j < 6;arr_j++){
            cin >> array[arr_i][arr_j];
        }
    }

    int max=-65355; //this holds down the maximum result
    int sum=0; //to store the sum of an individual hourglass

    int row=5; //includes 0 row and column
    int col=5;

    //to find the hourglass and calculate the sum of the hourglass
    for(int i=0; i<=row-2; i++)
        { for(int j=0; j<=col-2; j++) //makes all possible hourglasses withing individual rows
            { //following loops handles the individual logic to calculate the sum of an
hourglass as a whole entity
                sum=0;
                for(int k=0; k<2; k++) //represents the uppper and lower rows of the hourglass
                    { for(int l=0; l<3; l++)
                        {sum+=array[i+(2*k)][j+l];}
                    }
                //represents the single element of the hourglass
                sum+=array[i+1][j+1];
                // cout<<"\n sum---->"<<sum;
                if( sum>max)
                    {max=sum;}
            }
        }
    //cout<<"the maximum sum possible by an hourglass is "<<max;
    cout<<max;
    return 0;

}

```