Nama : Rizki Ramadhan
Nim : 20220040111
Kelas : TI22 F

1. Algorithms

Problem | Submissions | Leaderboard | Discussions | Editorial

Given an array of integers, find the sum of its elements.

For example, if the array $ar = [1, 2, 3]$, $1 + 2 + 3 = 6$, so return 6.

Function Description

Complete the simpleArraySum function in the editor below. It must return the sum of the array elements as an integer.

simpleArraySum has the following parameter(s):

- ar: an array of integers

Input Format

The first line contains an integer, n, denoting the size of the array.

The second line contains n space-separated integers representing the array's elements.

Constraints

$0 < n, ar[i] \leq 1000$

Output Format

Print the sum of the array's elements as a single integer.

Sample Input

```
6
1 2 3 4 10 11
```

Sample Output

```
31
```

Explanation

We print the sum of the array's elements: $1 + 2 + 3 + 4 + 10 + 11 = 31$.

```cpp
#include <bits/stdc++.h>
using namespace std;

/*
 * Complete the 'simpleArraySum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts vector<int> ar as parameter.
 */


int simpleArraySum(vector<int> ar) {
    int sum = 0;
    for (int i = 0; i < ar.size(); i++) {
        sum += ar[i];
    }
    return sum;
}

int main() {
    int n;
    cin >> n; // Read the size of the array

    vector<int> ar(n);
    for (int i = 0; i < n; i++) {
        cin >> ar[i]; // Read the array elements
    }

    int result = simpleArraySum(ar);
    cout << result << endl; // Print the sum of the array elements

    return 0;
}

```

2. Datastrutur

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:

```
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
 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
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:

```
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:

```cpp
#include <bits/stdc++.h>
using namespace std;

/*
 * Complete the 'hourglassSum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts 2D_INTEGER_ARRAY arr as parameter.
 */

#include <iostream>
#include <vector>
using namespace std;

int hourglassSum(vector<vector<int>>& arr) {
    int maxSum = INT_MIN;
    for (int i = 0; i <= 3; i++) {
        for (int j = 0; j <= 3; j++) {
            int sum = 0;
            // Top part of the hourglass
            sum += arr[i][j] + arr[i][j + 1] + arr[i][j + 2];
            // Middle part of the hourglass
            sum += arr[i + 1][j + 1];
            // Bottom part of the hourglass
            sum += arr[i + 2][j] + arr[i + 2][j + 1] + arr[i + 2][j + 2];

            maxSum = max(maxSum, sum);
        }
    }
    return maxSum;
}

int main() {
    vector<vector<int>> arr(6, vector<int>(6));

    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < 6; j++) {
            cin >> arr[i][j]; // Read the array elements
        }
    }

    int result = hourglassSum(arr);
    cout << result << endl; // Print the maximum hourglass sum

    return 0;
}

```

The hourglass with the maximum sum (19) is:

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