



# Equal Stacks 2

Problem

Submissions

Leaderboard

Discussions

You have three stacks of cylinders where each cylinder has the same diameter, but they may vary in height. You can change the height of a stack by removing and discarding its topmost (starting from index 0) cylinder any number of times. Find the maximum possible height of the stacks such that all of the stacks are exactly the same height. This means you must remove zero or more cylinders from the top of zero or more of the three stacks until they are all the same height, then return the height.

Example:

$h1 = [1, 2, 1, 1]$

$h2 = [1, 1, 2]$

$h3 = [1, 1]$

There are 4, 3 and 2 cylinders in the three stacks, with their heights in the three arrays. Remove the top 2 cylinders from  $h1$  (heights =  $[1, 2]$ ) and from  $h2$  (heights =  $[1, 1]$ ) so that the three stacks all are 2 units tall. Return it as the answer. Note: An empty stack is still a stack.

## Function Description

`int equalStacks(int h1_count, int* h1, int h2_count, int* h2, int h3_count, int* h3);`

Strictly use the above function prototype.

## Input Format

The first line contains three space-separated integers,  $n1$ ,  $n2$ , and  $n3$ , the numbers of cylinders in stacks 1, 2 and 3. The subsequent lines describe the respective heights of each cylinder in a stack from top to bottom:

The second line contains  $n1$  space-separated integers, the cylinder heights in stack. The first element is the top cylinder of the stack.

The third line contains  $n2$  space-separated integers, the cylinder heights in stack. The first element is the top cylinder of the stack.

The fourth line contains  $n3$  space-separated integers, the cylinder heights in stack. The first element is the top cylinder of the stack.

## Constraints

$0 < n1, n2, n3 \leq 105$

$0 < \text{height of any cylinder} \leq 100$

## Output Format

int: the height of the stacks when they are equalized

## Sample Input 0

```
5 3 4
3 2 1 1 1
4 3 2
1 1 4 1
```

## Sample Output 0

5



Contest ends in 15 minutes

Submissions: 53

Max Score: 10

Difficulty: Medium

Rate This Challenge:

[More](#)

C



```
1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5
6 int equalStacks(int h1_count, int* h1, int h2_count, int* h2, int h3_count, int* h3, int * sum);
7
8 int main() {
9
10     int h1, h2, h3;
11     scanf("%d %d %d", &h1, &h2, &h3);
12     int s1[h1], s2[h2], s3[h3], i;
13     int sum[3] = {0, 0, 0};
14
15     for(i = 0; i < h1; i++) {
16         scanf("%d", &s1[i]);
17         sum[0] += s1[i];
18     }
19     for(i = 0; i < h2; i++){
20         scanf("%d", &s2[i]);
21         sum[1] += s2[i];
22     }
23     for(i = 0; i < h3; i++){
24         scanf("%d", &s3[i]);
25         sum[2] += s3[i];
26     }
27
28     if(h1 == 0 || h2 == 0 || h3 == 0){
29         printf("0");
30         return 0;
31     }
32
33     equalStacks(h1, s1, h2, s2, h3, s3, sum);
34
35     return 0;
36 }
37
38 int equalStacks(int h1_count, int* h1, int h2_count, int* h2, int h3_count, int* h3, int *sum){
39     int idx1 = 0, idx2 = 0, idx3 = 0;
40
41     while(1) {
42         if (idx1 == h1_count || idx2 == h2_count || idx3 == h3_count) {
43             printf("0");
44             return 0;
45         }
46
47         if(sum[0] == sum[1] && sum[1] == sum[2]){
48             printf("%d", sum[0]);
49             return 0;
50         }
51
52         if(sum[0] >= sum[1] && sum[0] >= sum[2]){
53             sum[0] -= h1[idx1];
54             idx1++;
55         } else if(sum[1] >= sum[0] && sum[1] >= sum[2]){
56             sum[1] -= h2[idx2];
57             idx2++;
58         } else {
59             sum[2] -= h3[idx3];
60             idx3++;
61         }
62     }
63 }
```

Line: 1 Col: 1

 [Upload Code as File](#)☐ **Test against custom input**[Run Code](#)[Submit Code](#)

---

[Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |