

Worksheet - 2

**Student Name:** ANIKET KUMAR **UID:**20BCS5306

**Branch:** BE-CSE **Semester:** 5th

**Subject Name:** Competitive Coding **Section/Group:** 20BCS\_WM-703/B

* + 1. **AIM:**

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 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.

# Code:

**#include <**stdio.h**> #include <**string.h**> #include <**math.h**> #include <**stdlib.h**>**

**int** main() {

**int** n1,n2,n3,i,j=0,k=0,s1=0,s2=0,s3=0; scanf("%d %d %d",&n1,&n2,&n3);

**int** arr1[n1]; **int** arr2[n2]; **int** arr3[n3];

**for**(i=0;i<n1;i++){ scanf("%d",&arr1[i]); s1+=arr1[i];

}

**for**(i=0;i<n2;i++){ scanf("%d",&arr2[i]); s2+=arr2[i];

}

**for**(i=0;i<n3;i++){





scanf("%d",&arr3[i]); s3+=arr3[i];

} i=0;

**while**(1){

**if**((s1==s2 && s2==s3) || s1==0 || s2==0 || s3==0)

**break**; **if**(s1>=s2 && s1>=s3)

s1-=arr1[i++];

**else if**(s2>=s1 && s2>=s3) s2-=arr2[j++];

**else**

s3-=arr3[k++];

}

**if**(s1==0 || s2==0 || s3==0) printf("0");

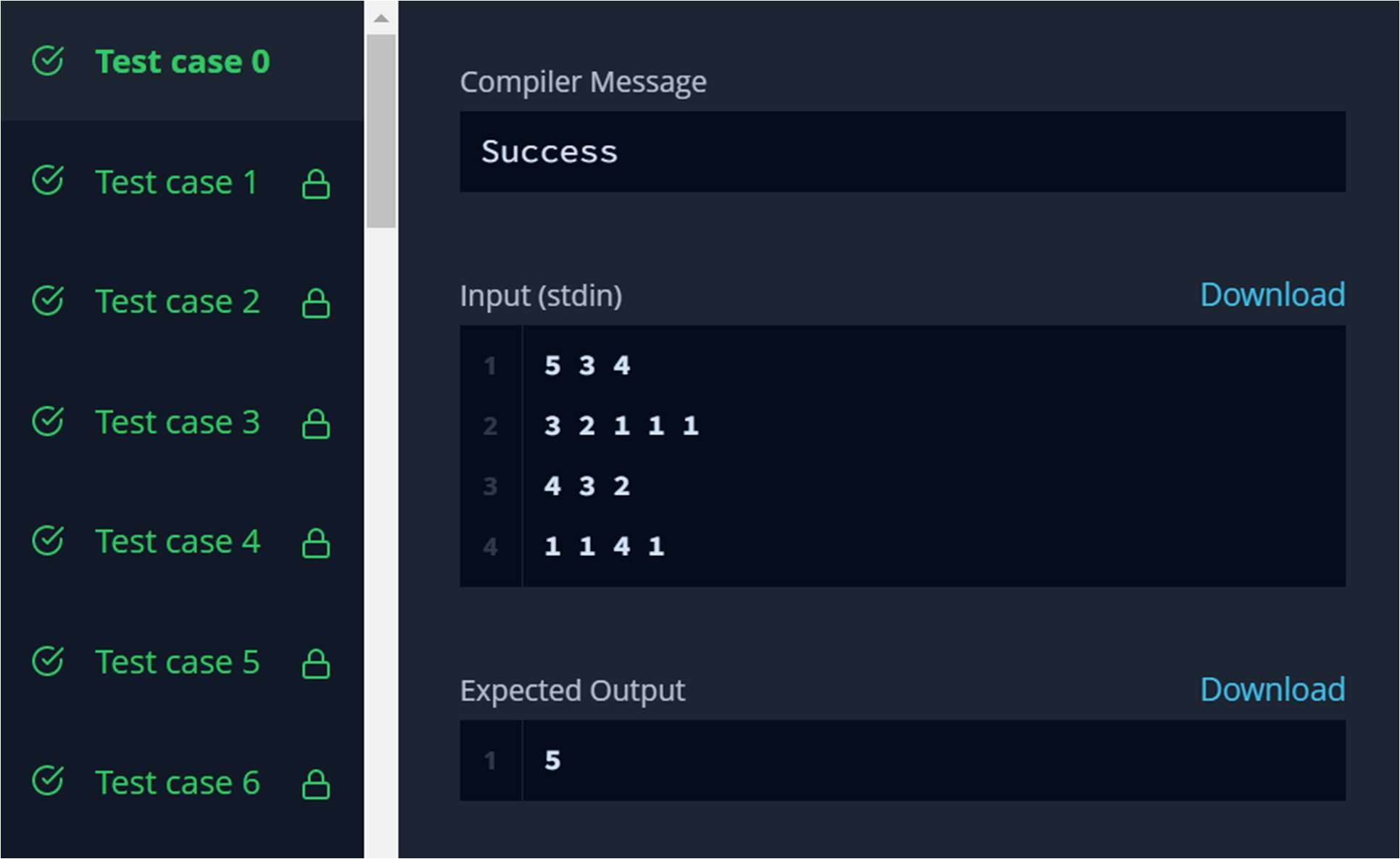
**else**

printf("%d",s1);

**return** 0;

}

**Hacker Rank Test Case / Output:**





# AIM:

You are given Q queries. Each query consists of a single number N. You can perform any of the 2 operations on N in each move:

1: If we take 2 integers a and b where N = a\*b (a!+1,b!=1) , then we can change N = max(a,b) 2: Decrease the value of N by 1.

Determine the minimum number of moves required to reduce the value of N to 0.

# Code:

**#include <**math.h**> #include <**stdio.h**> #include <**string.h**> #include <**stdlib.h**> #include <**assert.h**> #include <**limits.h**> #include <**stdbool.h**>**

**#define** N\_MAX 1000001

**int** solns[N\_MAX];

**void** initialize\_solns() {

**for** (**int** i = 0; i < N\_MAX; i++) { solns[i] = 0;

}

solns[1] = 1;

solns[2] = 2;

solns[3] = 3;

solns[4] = 3;

**for** (**int** i = 1; i < N\_MAX; i++) {

**if** (!solns[i] || solns[i-1] + 1 < solns[i]) { solns[i] = solns[i-1] + 1;

}

**for** (**int** j = 1; j <= i && j \* i < N\_MAX; j++) {

**if** (!solns[j\*i] || solns[i] + 1 < solns[j\*i]) { solns[j\*i] = solns[i] + 1;



}

}

}

}

**int** main() {

initialize\_solns();

**int** Q; scanf("%i", &Q);

**for**(**int** a0 = 0; a0 < Q; a0++){

**int** N; scanf("%i", &N);

printf("%d\n", solns[N]);

}

**return** 0;

}

**Hacker Rank Test Case / Output:**

