



Experiment Title-2

Student Name: SANSKAR AGRAWAL UID: 20BCS5914

Branch: CSE Section/Group: 806/B Semester: 5th Subject Name: CC Lab

Question 1:

1. Aim/Overview of the practical: Game of Two Stack

2. Task to be done/ Which logistics used:

Alexa has two stacks of non-negative integers, stack a[n] and stack b[m]where index 0 denotes the top of the stack. Alexa challenges Nick to play the following game:

- In each move, Nick can remove one integer from the top of either stack a or stack b.
- Nick keeps a running sum of the integers he removes from the two stacks.
- Nick is disqualified from the game if, at any point, his running sum becomes greater than some integer maxSum given at the beginning of the game.
- Nick's *final score* is the total number of integers he has removed from the two stacks.

Given ab, , and maxSum for g games, find the maximum possible score Nick can achieve.

3. Steps for experiment/practical/Code:

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

int main(){





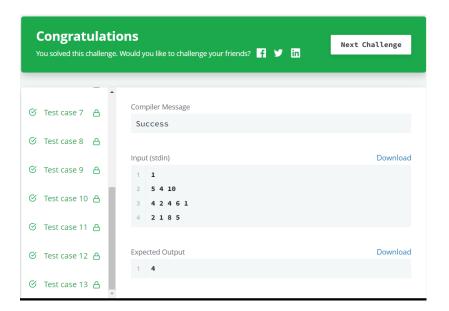


```
int g;
cin >> g;
for(int a0 = 0; a0 < g; a0++){
  int n;
  int m;
  int x;
  cin >> n >> m >> x;
  vector<int> a(n);
  for(int a_i = 0; a_i < n; a_{i++}){
    cin >> a[a_i];
  }
  vector<int> b(m);
  for(int b_i = 0; b_i < m; b_i + +)
    cin \gg b[b_i];
  int sum=0,count=0,temp=0,i=0,j=0;
  while(i < n && sum + a[i] <= x){
     sum+=a[i];
     i++;
  count=i;
  while(j < m \&\& i > = 0){
     sum+=b[j];
     j++;
     while(sum>x && i>0){
       i--;
       sum-=a[i];
     if(sum<=x && i+j>count)
       count=i+j;
  }
  cout<<count<<endl;</pre>
return 0;}
```





Result/Output/Writing Summary:









Ouestion 2:

- 1. Aim/Overview of the practical: Down to Zero II
- **2. Task to be done/ Which logistics used:** 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 $N = a \times b(a=1, b=1, b=1, b=1)$, 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.

3. Steps for experiment/practical/Code:

#include <bits/stdc++.h>

```
using namespace std;

int main() {
    int test;
    cin >> test;
    while (test--){
        int n;
        cin >> n;
        int steps = 0;
        if (n==0){
            cout << 0 << endl;
            continue;
        }
        if (n==1){
            cout << 1 << endl;
            continue;
        }vector<int> dist(n+1,0);
```





```
queue<int>q;
  q.push(n);
  dist[n] = 1;
  while (1){
     int element = q.front();
     q.pop();
     if(element == 2)
       cout << dist[2] + 1 << endl;
       break;
     if (dist[element-1] == 0)
       dist [element-1] = dist[element]+1;
       q.push(element-1);
     for (int i=2; i*i \le element; i++){
       if (element%i == 0){
          int maxfrac = element/i;
          if (dist[maxfrac] == 0) dist [maxfrac] = dist[element] + 1, q.push(maxfrac);
     }
return 0;
```







4. Result/Output/Writing Summary:

