Experiment 2

Student Name: ABHAY KUMAR
UID: 20BCS9222
Branch: BE-CSE
Section/Group:616-A
Semester: 5
Date of Performance:26-08-2022
Subject Name: Competitive Coding

1. Aim: A matching pair of brackets is *not balanced* if the set of brackets it encloses are not matched. For example, `{[(])}` is not balanced because the contents in between `{` and `}` are not balanced. The pair of square brackets encloses a single, unbalanced opening bracket, `(`, and the pair of parentheses encloses a single, unbalanced closing square bracket, `]`.

By this logic, we say a sequence of brackets is *balanced* if the following conditions are met:

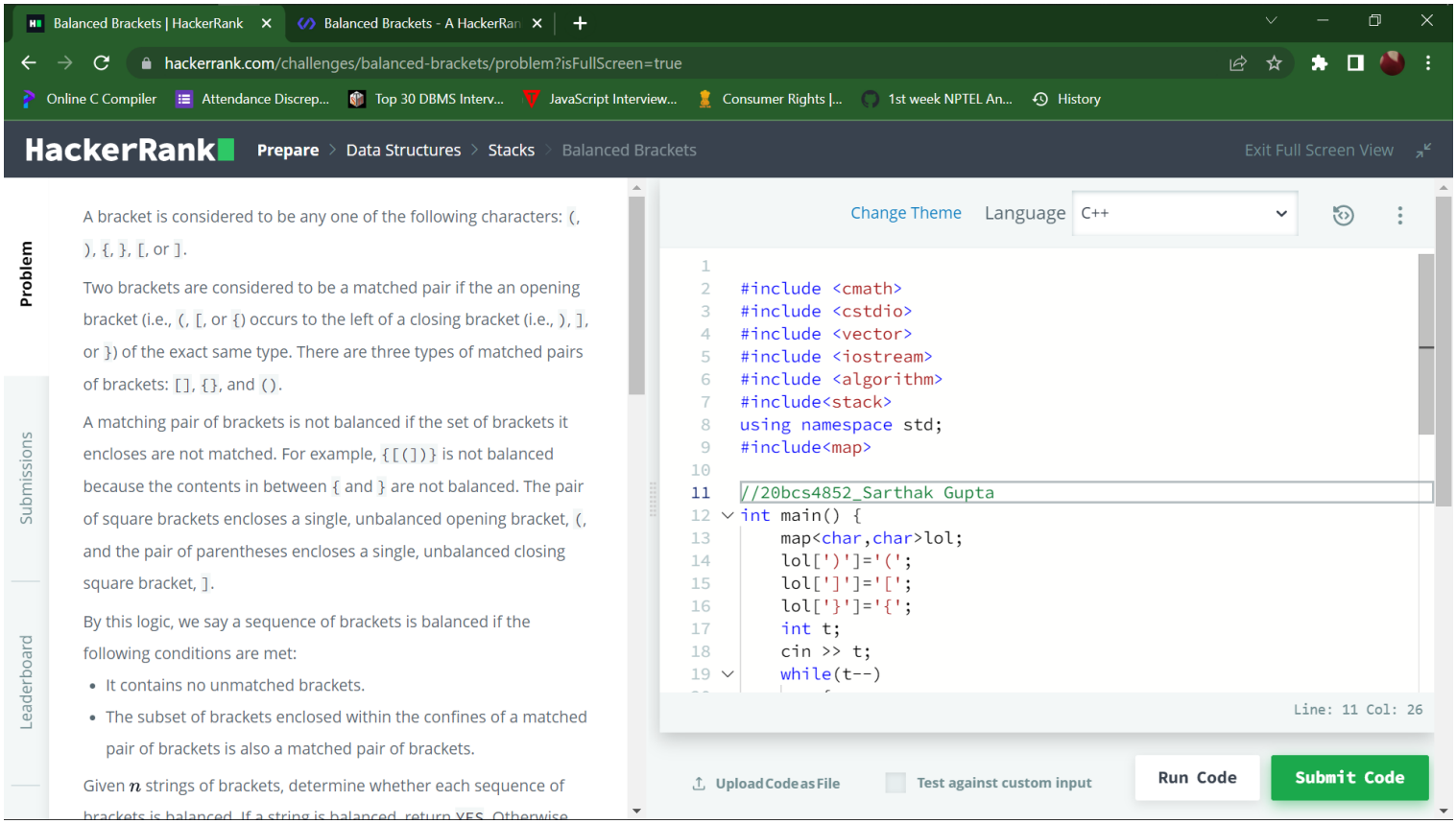
- **It contains no unmatched brackets.**
- **The subset of brackets enclosed within the confines of a matched pair of brackets is also a matched pair of brackets.**

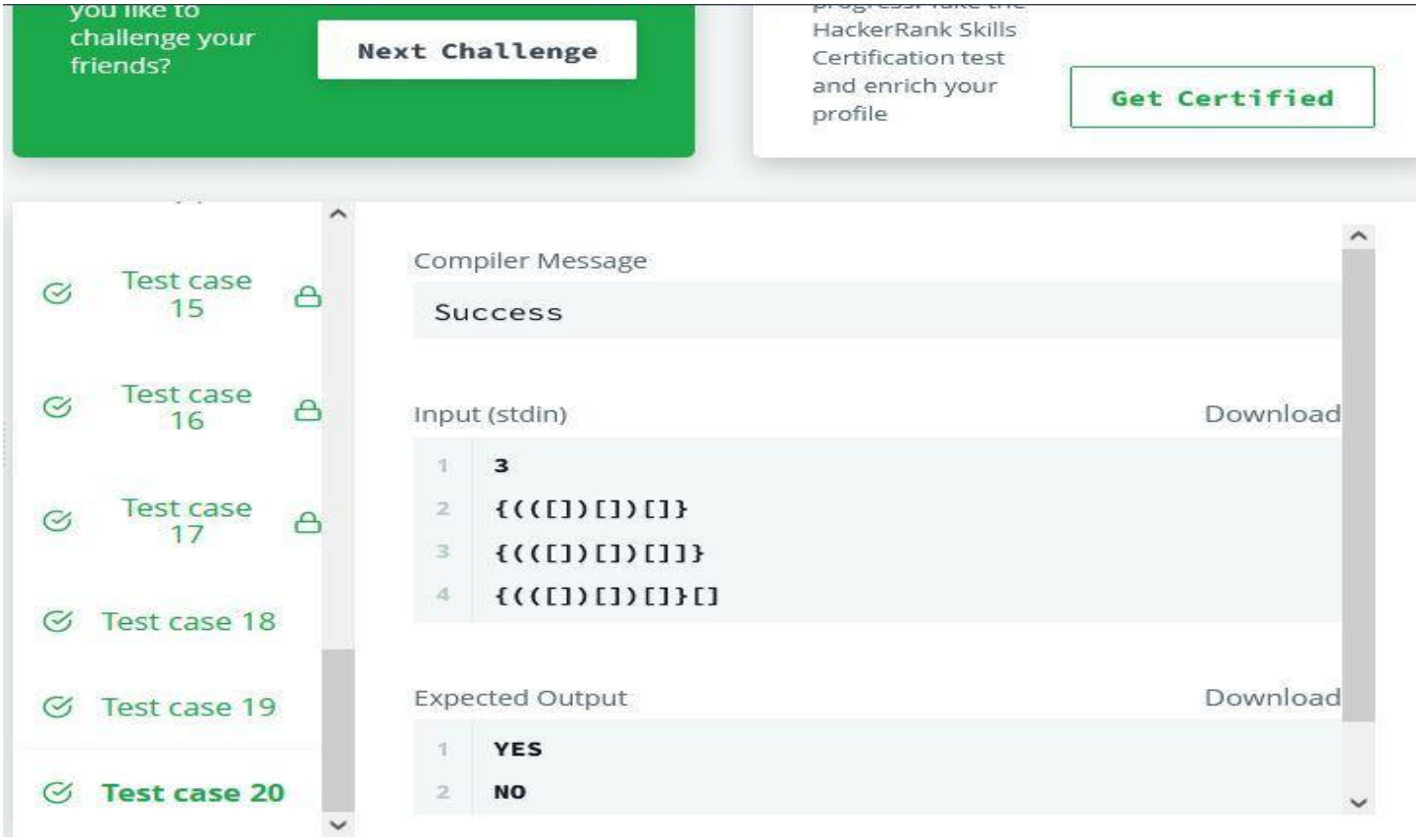
CODE:

```
#include <bits/stdc++.h>
using namespace std;
int main(){
   //SARTHAK GUPTA(20BCS4852)
   int t;
   cin >> t;
   while(t--){
      string str;
      cin >> str;
      vector<char> v1;
      long long top=-1;
      for(int i=0;i<str.length();i++){
```

```
        if(str[i]=='{'||str[i]=='['||str[i]=='('||top == -1){
            v1.push_back(str[i]);
            top++;
        }
        else if((str[i]=='}' && v1[top]=='{')|| (str[i]==']' && v1[top]=='[')|| (str[i]==')'&&v1[top]=='(')){
            v1.pop_back();
            top--;
        }
        else {
            v1.push_back(str[i]);
            top++;
        }
    }
    if(v1.size() == 0){
        cout << "YES" << endl;
    }else cout << "NO" << endl;
  }
}
```

OUTPUT:


Balanced Brackets | HackerRank
Balanced Brackets - A HackerRan
hackerrank.com/challenges/balanced-brackets/problem?isFullScreen=true
Online C Compiler
Attendance Discrep...
Top 30 DBMS Interv...
JavaScript Interview...
Consumer Rights |...
1st week NPTEL An...
History
HackerRank
Prepare
Data Structures
Stacks
Balanced Brackets
Exit Full Screen View
Problem
Submissions
Leaderboard
A bracket is considered to be any one of the following characters: (,), {, }, [, or].
Two brackets are considered to be a matched pair if the an opening bracket (i.e., (, [, or {) occurs to the left of a closing bracket (i.e.,),], or }) of the exact same type. There are three types of matched pairs of brackets: [], {}, and ().
A matching pair of brackets is not balanced if the set of brackets it encloses are not matched. For example, {[(])} is not balanced because the contents in between { and } are not balanced. The pair of square brackets encloses a single, unbalanced opening bracket, (, and the pair of parentheses encloses a single, unbalanced closing square bracket,].
By this logic, we say a sequence of brackets is balanced if the following conditions are met:
It contains no unmatched brackets.
The subset of brackets enclosed within the confines of a matched pair of brackets is also a matched pair of brackets.
Given n strings of brackets, determine whether each sequence of
Change Theme
Language
C++
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
#include<stack>
using namespace std;
#include<map>
//20bcs4852_Sarthak Gupta
int main() {
map<char,char>lol;
lol[')']='(';
lol[']']='[';
lol['}']='{';
int t;
cin >> t;
while(t--)
Line: 11 Col: 26
Upload Code as File
Test against custom input
Run Code
Submit Code



2.

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

1: If we take 2 integers a and b where N=a*b then we can change n= amx(a,b) 2: Decrease the value of by 1

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

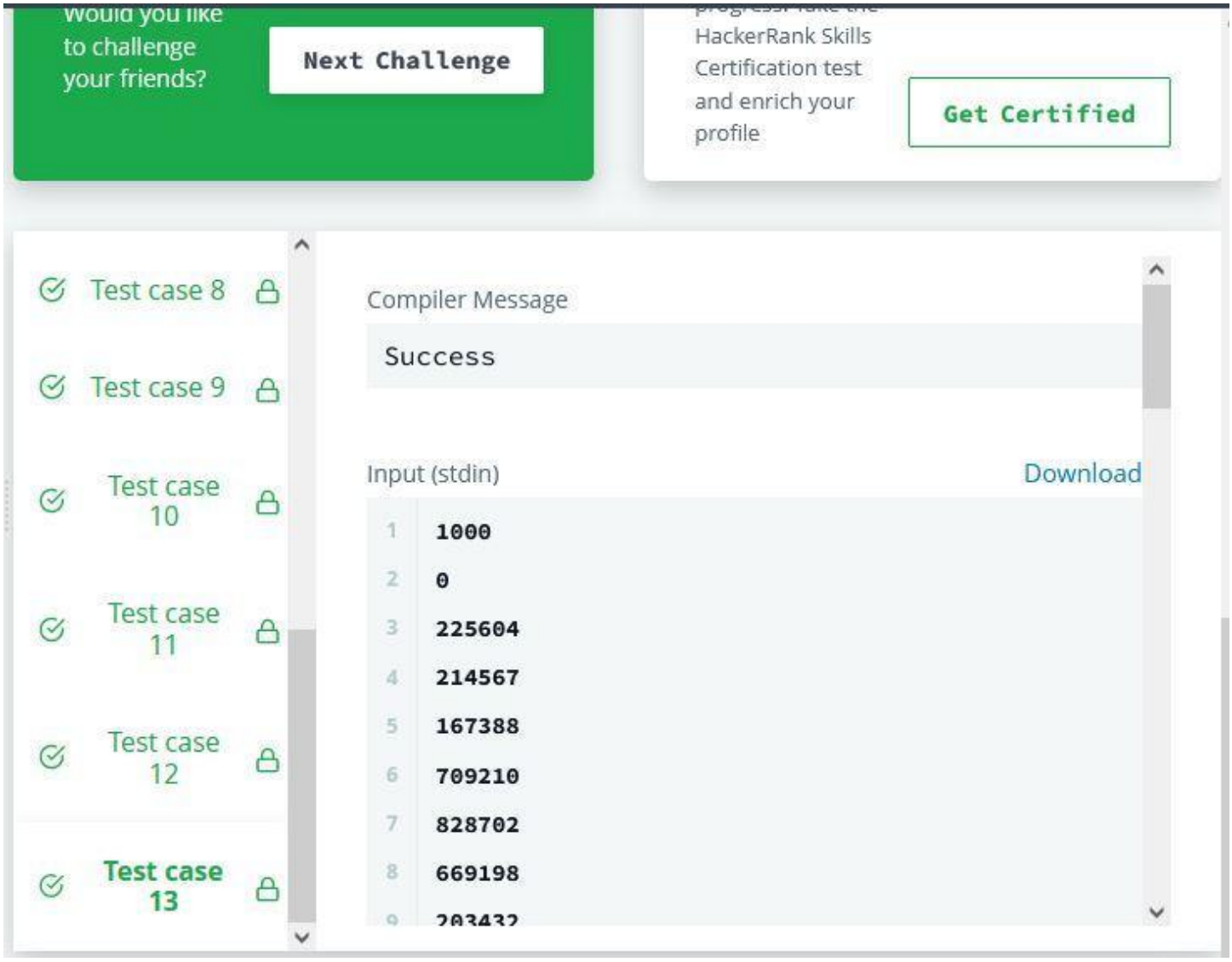
CODE:

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long
int main()
```

```
{
  //SARTHAK GUPTA(20BCS4852)
  ll t;
  cin>>t;
  while(t--)
  {
     ll a;
     cin>>a;
     queue<pair<ll,ll>>v;
     vector<ll>v2(1e6 + 1,0);
     v.push({a,0});
     while(v.size() > 0)
     {
        pair<ll,ll> z=v.front();
        v.pop();
        if(z.first==0)
        {
           break;
        }
        else{
           if(v2[z.first-1]==0)
           {
              v.push({z.first-1,z.second+1});
              v2[z.first-1]=z.second+1;
           }
           for(ll i=sqrt(z.first);i>1;i--)
           {
              if(z.first%i==0)
              {
                 if(v2[z.first/i]==0)
                 {
                    v.push({z.first/i,z.second+1});
                    v2[z.first/i]=z.second+1;
                 }
              }
           }
        }
     }
     cout<<v2[0]<<endl;
  }
```

}

OUTPUT:


would you like to challenge your friends?
Next Challenge
HackerRank Skills Certification test and enrich your profile
Get Certified
Test case 8
Test case 9
Test case 10
Test case 11
Test case 12
Test case 13
Compiler Message
Success
Input (stdin)
Download
1 1000
2 0
3 225604
4 214567
5 167388
6 709210
7 828702
8 669198