CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Computer Science & Engineering

Course No: CSE- 244

Course Title:
Algorithm Design & Analysis (Sessional)

Experiment No: 04

Name Of the Experiment: Greedy Technique.

Identity Details

Name of the Student: Md. Safayat Bin Nasir

Student Id Number: 2004099

Level: 2 Term: 2 Section: B

Group: B1

Date of performance: 01-10-2023

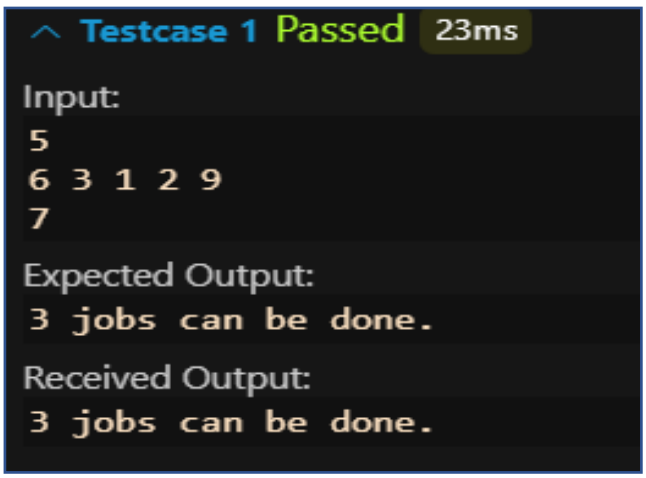
Date of Submission: 08-10-2023

Remarks

Program 1: Given an array of integers where each element indicates the time a job takes for completion. Calculate the maximum number of jobs you can do in the given period of time.

```cpp
#include <bits/stdc++.h>
using namespace std;
#define sad '\n'
#define all(b) b.begin(), b.end()
int main()
{
    int n ;
    cin >> n;
    vector<int>v(n);
    for(auto &x:v) cin >> x;
    sort(all(v));
    int time;
    cin >> time;
    int ans = 0 , ind  = -1;
    for(int i = 0 ; i < n ; i++){
        //cout  << v[i] << sad;
        ans += v[i];
        if(ans > time){
            ind = i;
            break;
        }
    }
    if(ind == -1) ind = n;
    cout << ind << " jobs can be done."<< sad;
}
```

Output:



Program 2: You will be given n jobs with their starting and ending time. Find maximum number of jobs that can be done by a single person.

```cpp
#include <bits/stdc++.h>
using namespace std;
#define sad '\n'
#define all(b) b.begin(), b.end()
bool comp(pair<int , int>&a , pair<int , int>&b){
    return (a.second < b.second || a.second == b.second and a.first < b.first);
}
int main()
{
    int n ;
    cin >> n;
    vector<pair<int ,int>>v;
    for(int i = 0  ; i < n ; i++){
        int a , b;
        cin >> a >> b;
        v.push_back({a , b});
    }
    sort(all(v)  , comp);
    int a = 0 , b = 0, ans = 0, cnt = 0;
    for(auto &x:v){
        if(x.first >= b){
            a = x.first, b = x.second;
            ans++;
        }
    }
    cout << ans << " jobs can be done."<< sad;
}
```

Output:

Testcase 1 Passed 31ms

Input:

```
3
10 20
20 30
12 25
```

Expected Output:

```
2 jobs can be done.
```

Received Output:

```
2 jobs can be done.
```