**Worksheet-4**

**Student Name:-** Pushpraj Roy **UID:-** 20BCS9866

**Branch:-** BE- CSE **Section/Group:-** WM\_617 “A”

**Subjetct Code:-** 20CSP-314 **Semester:-** 5th

**Subject Name:-** Competitive Coding Lab

**Problem 1:- Counting Sort**

<https://www.hackerrank.com/challenges/countingsort4/problem?isFullScreen=true>

**Code:-**

#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);

string rtrim(const string &);

vector<string> split(const string &);

void countSort(vector<vector<string>> arr)

{ int n=arr.size();

map<int, vector<string>>m;

for(int i = 0; i< (n/2); ++i)

arr[i][1] = "-";

for(int i = 0; i<n; ++i)

m[stoi(arr[i][0])].push\_back(arr[i][1]);

for(auto pair : m)

for( auto s : pair.second)

cout << s << " ";

}

int main()

{

string n\_temp;

getline(cin, n\_temp);

int n = stoi(ltrim(rtrim(n\_temp)));

vector<vector<string>> arr(n);

for (int i = 0; i < n; i++)

{ arr[i].resize(2);

string arr\_row\_temp\_temp;

getline(cin, arr\_row\_temp\_temp);

vector<string> arr\_row\_temp = split(rtrim(arr\_row\_temp\_temp));

for (int j = 0; j < 2; j++) {

string arr\_row\_item = arr\_row\_temp[j];

arr[i][j] = arr\_row\_item;

}

}

countSort(arr);

return 0;

}

string ltrim(const string &str)

{ string s(str);

s.erase(

s.begin(), find\_if(s.begin(), s.end(), not1(ptr\_fun<int, int>(isspace)))

);

return s;

}

string rtrim(const string &str)

{ string s(str);

s.erase(

find\_if(s.rbegin(), s.rend(), not1(ptr\_fun<int, int>(isspace))).base(), s.end()

);

return s;

}

vector<string> split(const string &str)

{ vector<string> tokens;

string::size\_type start = 0;

string::size\_type end = 0;

while ((end = str.find(" ", start)) != string::npos)

{ tokens.push\_back(str.substr(start, end - start));

start = end + 1;

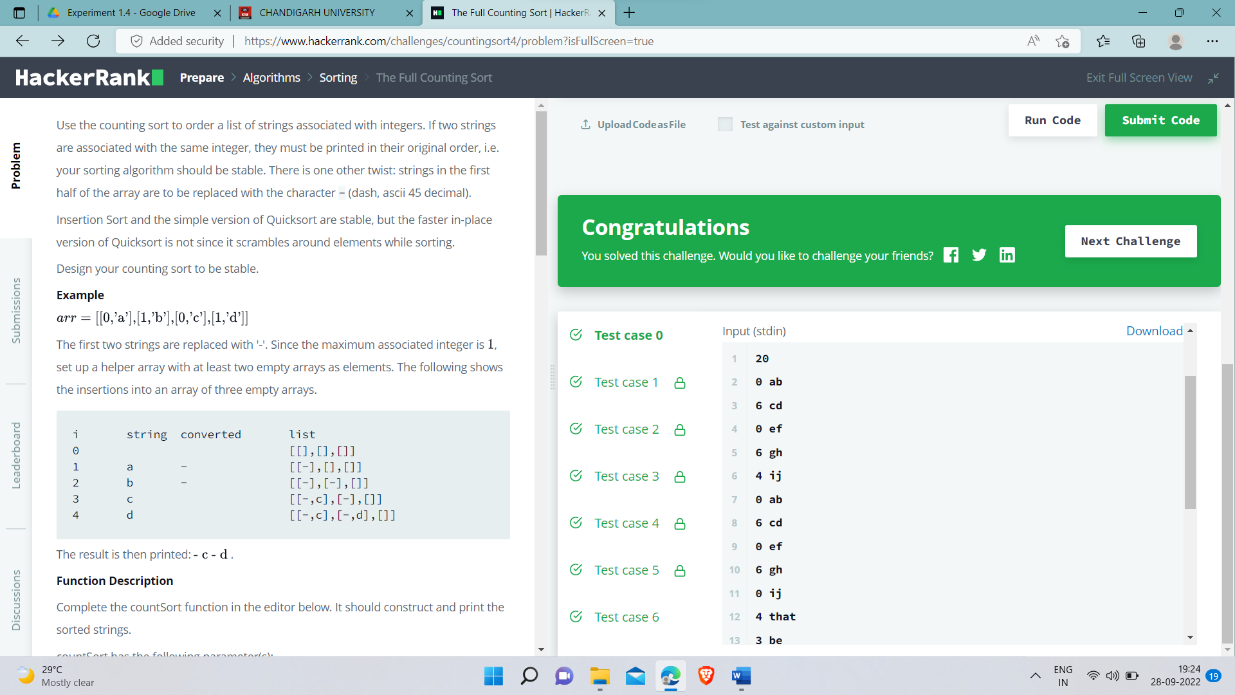
}

tokens.push\_back(str.substr(start));

return tokens;

}

**Output:-**



**Problem 2:- Insertion Sort**

<https://www.hackerrank.com/challenges/insertion>[-sort/problem?isFullScreen=true](https://www.hackerrank.com/challenges/insertion-sort/problem?isFullScreen=true)

**Code:-**

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

long long iNum = 0;

void cntInv(vector<int>& a, vector<int>& b, int l, int r){

if(r - l < 2)

return;

int mid = (l + r) / 2;

cntInv(b, a, l, mid);

cntInv(b, a, mid, r);

int i = mid - 1;

int j = r - 1;

int k = r - 1;

while(i >= l && j >= mid){

if(a[j] >= a[i])

b[k--] = a[j--];

else{

iNum += j - mid + 1;

b[k--] = a[i--];

}

}

while(i >= l){

b[k--] = a[i--];

}

while(j >= mid){

b[k--] = a[j--];

}

}

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

int T;

cin >> T;

while(T--){

vector<int> a;

int n;

cin >> n;

for(int i = 0; i < n; ++i){

int num;

cin >> num;

a.push\_back(num);

}

vector<int> b(a);

iNum = 0;

cntInv(a, b, 0, a.size());

cout << iNum << endl;

}

return 0;

}

**Output:-**

