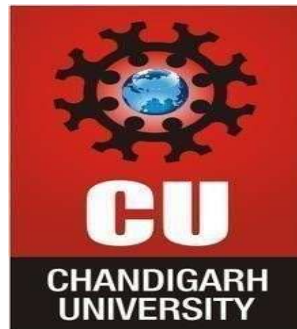


**CHANDIGARH
UNIVERSITYUNIVERSITYINSTITUTE OF EN
GINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



StudentName: Aniket Kumar

Branch: BE-CSE

SubjectName: Competitive Coding

UID: 20BCS5306

Semester: 5th

Section/Group: 20BCS_WM-703/B

LAB INDEX

| Sr. No | Program | Date | Evaluation | | | | Sign |
|-----------|---|------|------------|-----------|------------|---------------|------|
| | | | LW (12) | VV (8) | FW (10) | Total (30) | |
| 1. | Todemonstratetheconceptofarrays | | | | | | |
| 2. | Todemonstratetheconceptofstackand queues | | | | | | |
| 3. | Todemonstrate theconcept ofLinkedlist | | | | | | |
| 4. | Todemonstratetheconcept ofSearchingandsorting | | | | | | |
| 5. | Todemonstrate theconcept ofGraphs | | | | | | |
| 6. | Todemonstrate theconcept ofTrees | | | | | | |
| 7. | Todemonstrate theconcept ofStrings | | | | | | |
| 8. | Todemonstratetheconceptof Dynamicprogramming | | | | | | |
| 9. | TodemonstratetheconceptofBacktracking | | | | | | |
| 10. | To demonstrate theconcept ofGreedyand Branchand Bound | | | | | | |

10.1 AIM OF THE EXPERIMENT- Greedy and Branch and Bound

TASK TO BE DONE/LOGISTICS USED--

Marc loves cupcakes, but he also likes to stay fit. Each cupcake has a calorie count, and Marc can walk a distance to expend those calories. If Marc has eaten j cupcakes so far, after eating a cupcake with c calories he must walk at least $2^j \times c$ miles to maintain his weight.

<https://www.hackerrank.com/challenges/marcs-cakewalk/problem?isFullScreen=false>

CODE:(C++)

```
#include

<bits/stdc++.h>using namespace

space std;

string ltrim(const string
&);string rtrim(const string&);
vector<string>split(const string&);long mar

csCakewalk(vector<int> calorie){

    sort(calorie.begin(),calorie.end(),greater<int>());

    long sum=0;
    for(int i=0;i<calorie.size(); i++)
    {
        sum+=pow(2, i) *calorie[i];
```

```

    }

    return sum;
}

int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    string
    n_temp; getline(cin, n_
    temp);

    int n = stoi(ltrim(rtrim(n_temp)));

    string
    calorie_temp_temp; getline(cin, ca
    lorie_temp_temp);

    vector<string> calorie_temp =

    split(rtrim(calorie_temp_temp)); vector<int> calorie(n);

    for(int i=0; i<n; i++){
        int calorie_item = stoi(calorie_temp[i]);

        calorie[i] = calorie_item;
    }

    long result =

    marcsCakewalk(calorie); fout<<result<<

    "\n"; fout.close();

    return 0;
}

string ltrim(const string &str)
{
    strings(str);

    s.erase(
        s.begin(),
        find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
    );
}

```

```

    returns;
}

string rtrim(const string &str)
{strings(str);

    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int,
        int>(isspace))).base(),s.end()
    );

    returns;
}

vector<string>split(const string &str)
{vector<string>tokens;

    string::size_type start =
    0;string::size_typeend =0;

    while ((end = str.find(" ", start)) != string::npos)
        {tokens.push_back(str.substr(start,end-start));

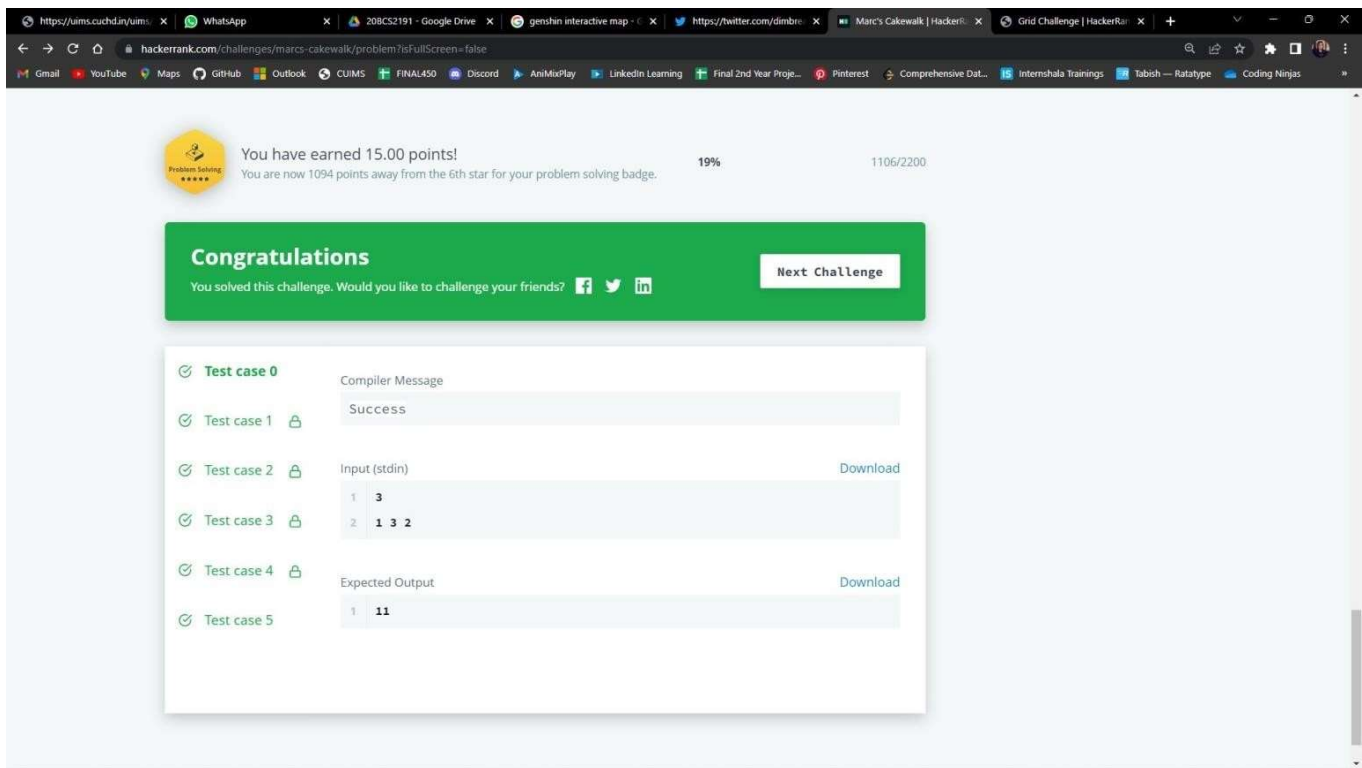
            start= end+1;
        }

    tokens.push_back(str.substr(start));

    returntokens;
}

```

OUTPUT:



10.2 AIM OF THE EXPERIMENT – Greedy and Branch and Bound

TASK TO BE DONE / LOGISTICS USED –

Given a square grid of characters in the range `ascii[a-z]`, rearrange elements of each row alphabetically, ascending. Determine if the columns are also in ascending alphabetical order, top to bottom. Return YES if they are or NO if they are not.

Example

`grid = ['abc', 'ade', 'efg']`

The grid is illustrated below.

```
a b c
a d e
e f g
```

The rows are already in alphabetical order. The columns `a a e`, `b d f` and `c e g` are also in alphabetical order, so the answer would be YES. Only elements within the same row can be rearranged. They cannot be moved to a different row.

<https://www.hackerrank.com/challenges/grid-challenge/problem?isFullScreen=false>

CODE:(INC++)

```
#include

<bits/stdc++.h>using namespace

std;

string ltrim(const string
&);string rtrim(const string&);

string gridChallenge(vector<string> grid)
{
    for(int i=0; i<grid.size(); i++){
        sort(grid[i].begin(), grid[i].end());
    }
    for(int i=1; i<grid.size(); i++){
        for(int j=0; j<grid.size(); j++){
            if(grid[i-1][j]>grid[i][j])
                {return "NO";
            }
        }
    }
    return "YES";
}

int main()
{
    ofstream fout(getenv("OUTPUT_PATH"));

    string
    t_temp;getline(cin, t_
temp);

    int t=stoi(ltrim(rtrim(t_temp)));

    for (int t_itr = 0; t_itr< t; t_itr++)
        {string n_temp;
        getline(cin, n_temp);
```

```

    int n = stoi(ltrim(rtrim(n_temp))); vec
    tor<string>grid(n);

    for (int i = 0; i < n; i++)
        {stringgrid_item;getline(
            cin,grid_item);

            grid[i]=grid_item;
        }

    stringresult=gridChallenge(grid);

    fout<<result<<"\n";
}

fout.close();

return 0;
}

string ltrim(const string &str)
{strings(str);

    s.erase(
        s.begin(),
        find_if(s.begin(),s.end(),not1(ptr_fun<int,int>(isspace)))
    );

    return s;
}

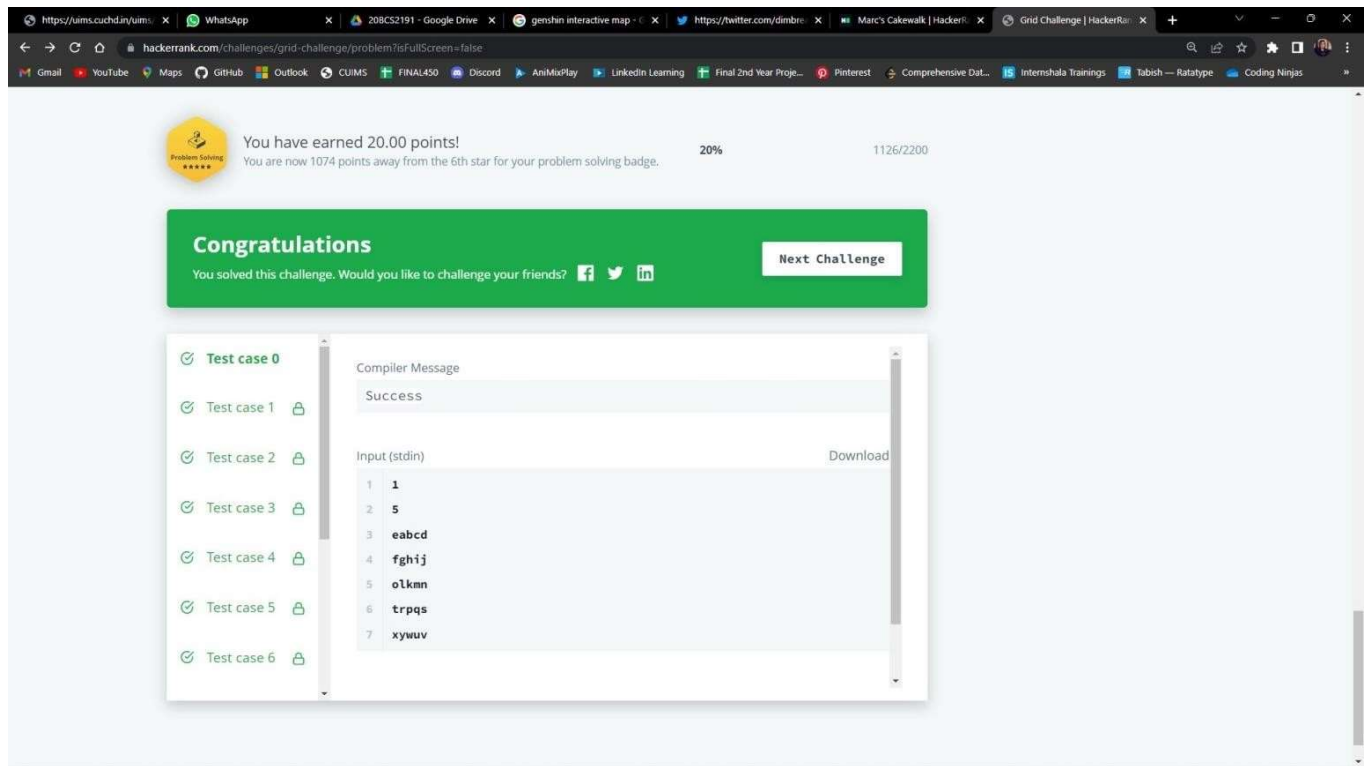
string rtrim(const string &str)
{strings(str);

    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int,
        int>(isspace))).base(),s.end()
    );

    return s;
}

```


OUTPUT:



Learning Outcomes (What I have learnt):

- It will provide the modest experience that allows students to develop and improve their experimental skills and develop ability to analyze data.
- Ability to demonstrate the practical skill on measurements and instrumentation techniques of some Physics experiments. Students will develop the ability to use appropriate physical concepts to obtain quantitative solutions to problems in physics.
- Students will demonstrate basic experimental skills by setting up laboratory equipment safely and efficiently, plan and carry out experimental procedures, and report verbally and in written language the results of the experiment.
- Students will develop skills by the practice of setting up and conducting an experiment with due regard to minimizing measurement error.