



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 3.3

Student Name: Jitesh Kumar

UID: 20BCS1371

Branch: BE-CSE

Section/Group: WM_903-A

Semester: 5

Subject Code: 20CSP-317 Subject

Name: Competitive Coding

Marc's Cake Walk

Aim: 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 cupcakes so far, after eating a cupcake with calories he must walk *at least* miles to maintain his weight. You are given a function,

CODE:

//Jitesh Kumar

```
import java.util.Arrays;
import java.util.Scanner;

class Solution {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int arr[] = new int[n];
        for (int i = 0; i < n; i++)
            arr[i] = sc.nextInt();
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
Arrays.sort(arr);
long res = 0;
for (int i = 0; i < n; i++) {
    res += (long) (Math.pow(2, i) * arr[n - i - 1]);
}
System.out.println(res);
sc.close();
}
}
```

OUTPUT:

Congratulations

You solved this challenge. Would you like to challenge your friends?

[f](#) [t](#) [in](#)

[Next Challenge](#)

✓ Test case 0	Compiler Message				
✓ Test case 1 🔒	Success				
✓ Test case 2 🔒	Input (stdin) Download				
✓ Test case 3 🔒	<table><tr><td>1</td><td>3</td></tr><tr><td>2</td><td>1 3 2</td></tr></table>	1	3	2	1 3 2
1	3				
2	1 3 2				
✓ Test case 4 🔒	Expected Output Download				
✓ Test case 5	<table><tr><td>1</td><td>11</td></tr></table>	1	11		
1	11				

Grid Challenge

Aim: 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.:

CODE:

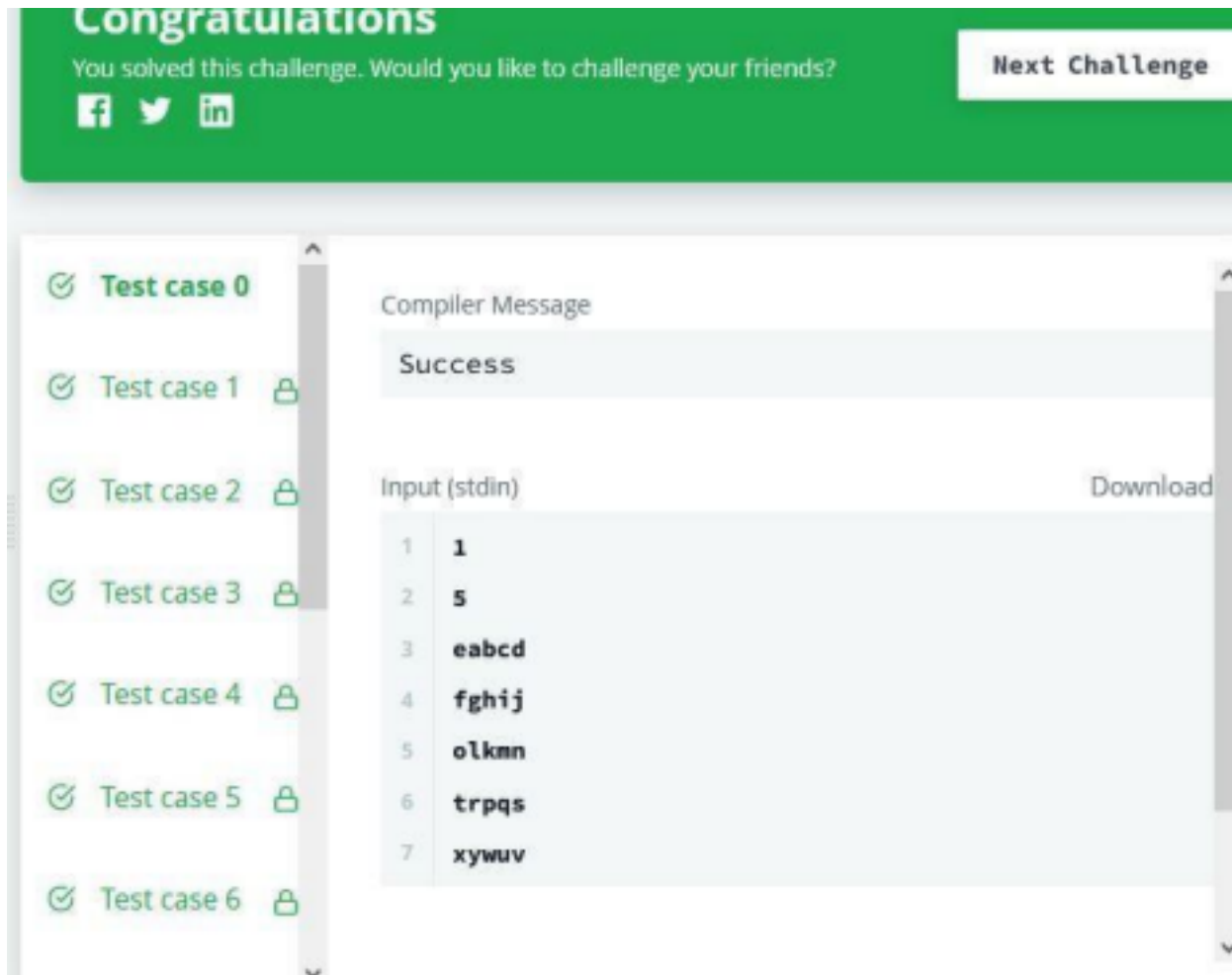
```
#include <iostream>
#include <algorithm>
#include <string>
using namespace std;

string s[111];
//Jitesh

int main() {
    int t;
    cin >> t;
    while (t--) {
        int n;
        cin >> n;
        for (int i = 0; i < n; i++) cin >> s[i], sort(s[i].begin(),
s[i].end());
        bool flag = true;
        for (int i = 0; i < n;
i++) for (int j = 0; j + 1
< n; j++) if (s[j][i] >
s[j + 1][i]) flag = false;
        puts(flag ? "YES" : "NO");
    }
    return 0;
}
```

}

OUTPUT:



Learning Outcomes:

1. Learn About the Greedy Approach
2. Learn About How to Use Greedy Approach in different questions