



Experiment 3.3

Student Name: Nishant Kumar Mehta

UID: 21BCS3402

Branch: BE-CSE

Section/Group: IoT-602/B

Semester: 5

Date of Performance: 26-10-2023

Subject Name: Advance Programming Lab

Subject Code: 21CSP-314

1. Aim: Implement the problems based on Branch and Bound

2.Objective:

1. 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.

2. 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

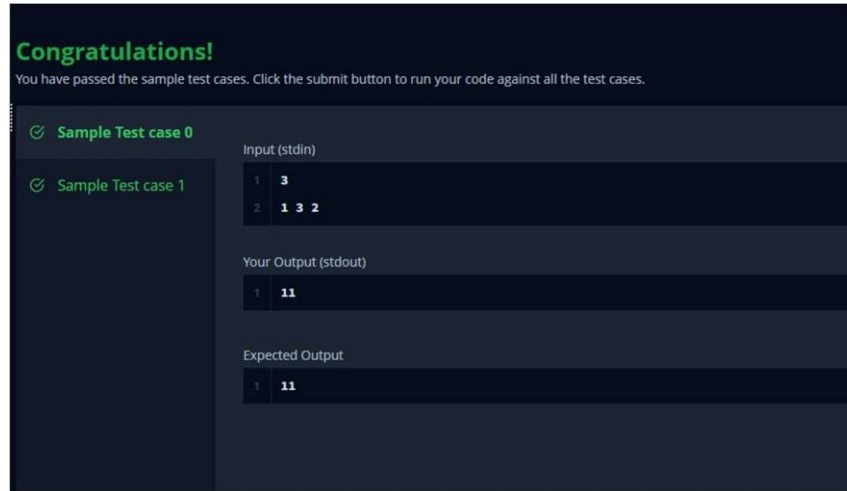
3. Script and Output:

Program 1:-

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int[] calories = new int[n];
        for(int calories_i=0; calories_i < n; calories_i++){
            calories[calories_i] = in.nextInt();
        }
        Arrays.sort(calories);
        long ans = 0;
        for (int i = 0; i < n; i++) {
            ans += ((long)calories[n-i-1])<<i;
        }
        System.out.println(ans);
    }
}
```

Output:-



Program 2:-

```
import java.util;
class Result {
    public static String gridChallenge(List<String> grid) {
        char[][] arr = new char[grid.size()][grid.get(0).length()];
        for(int i=0;i<grid.size(); i++){
            char[] word = grid.get(i).toCharArray();
            Arrays.sort(word);
            for(int j=0; j<word.length; j++){
                arr[i][j] = word[j];
            }
        }
        for(int col=0; col<arr[0].length; col++){
            for(int row=0; row<grid.size()-1; row++){
                if(arr[row][col] > arr[row+1][col]){
                    return "NO";
                }
            }
        }
        return "YES";
    }
}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new
        FileWriter(System.getenv("OUTPUT_PATH")));

        int t = Integer.parseInt(bufferedReader.readLine().trim());

        IntStream.range(0, t).forEach(tItr -> {
            try {
                int n = Integer.parseInt(bufferedReader.readLine().trim());
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
List<String> grid = IntStream.range(0, n).mapToObj(i -> {
    try {
        return bufferedReader.readLine();
    } catch (IOException ex) {
        throw new RuntimeException(ex);
    }
})
.collect(toList());
String result = Result.gridChallenge(grid);
bufferedWriter.write(result);
bufferedWriter.newLine();
} catch (IOException ex) {
    throw new RuntimeException(ex);
}
});
bufferedReader.close();
bufferedWriter.close();
}
```

Output:-

