

**UNIVERSITY INSTITUTE OF ENGINEERING**

**Department of Computer Science & Engineering**

**Subject Name:** Competitive Coding

**Subject Code:** 20CSP-314

**Submitted to: Submitted by:**

Er. Mamta Punia Name: Sahil Kaundal

UID: 21BCS8197

Section: 616

Group: A

**INDEX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ex. No** | **List of Experiments** | **Conduct (MM: 12)** | **Viva**  **(MM: 10)** | **Record (MM: 8)** | **Total**  **(MM: 30)** | **Remarks/Signature** |
| 1 | To demonstrate the concept of Array. |  |  |  |  |  |
| 2 | To demonstrate the concept of Stack and Queue. |  |  |  |  |  |
| 3 | To demonstrate the concept of Linked List. |  |  |  |  |  |
| 4 | Sorting and Searching: Implement the concept of Searching and Sorting techniques. |  |  |  |  |  |
| 5 | To implement the concept of Graphs. |  |  |  |  |  |
| 6. | To demonstrate the concept of Tree Data Structure |  |  |  |  |  |
| 7. | To Demonstrate the concept of String Data Structure |  |  |  |  |  |
| 8. | Dynamic Programming |  |  |  |  |  |
| 9. | Backtracking |  |  |  |  |  |
| 10. | Branch and Bound, Greedy |  |  |  |  |  |

**Experiment 10.1**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** BE CSE (Lateral Entry) **Section/Group:** 616/A

**Semester:** 5th **Date of Performance:** 10/11/2022

**Subject Name:** CC Lab **Subject Code:** 20CSP-314

1. **Aim/Overview of the practical:**

Branch and Bound, Greedy

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.

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

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

1. **Objective:**
   * To understand the concept of Branch and Bound.
   * To implement the concept of Greedy.

**4. Code:**

#include <math.h>

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <assert.h>

#include <limits.h>

#include <stdbool.h>

void swap(int \*a,int \*b)

{

    int temp;

    temp = \*a;

    \*a = \*b;

    \*b=temp;

}

int partition(int \*x,int start,int end)

{

    int pivot,pindex,i;

    pivot = x[end];

    pindex = start;

    for(i=start;i<end;i++)

    {

        if(x[i]>=pivot)

          {

            swap(&x[i],&x[pindex]);

            pindex = pindex + 1;

          }

    }

    swap(&x[pindex],&x[end]);

    return pindex;

}

void quicksort(int \*x,int start,int end)

{

    if(start<end)

   {

     int i = partition(x,start,end);

      quicksort(x,start,i-1);

      quicksort(x,i+1,end);

    }

}

int main(){

    int n;

    scanf("%d",&n);

    int \*calories = malloc(sizeof(int) \* n);

    for(int calories\_i = 0; calories\_i < n; calories\_i++)

    {

       scanf("%d",&calories[calories\_i]);

    }

    quicksort(calories,0,n-1);

    int i;

    long long int sum = 0;

    for(i=0;i<n;i++)

    {

        sum += calories[i]\*((long long int)pow(2,i));

    }

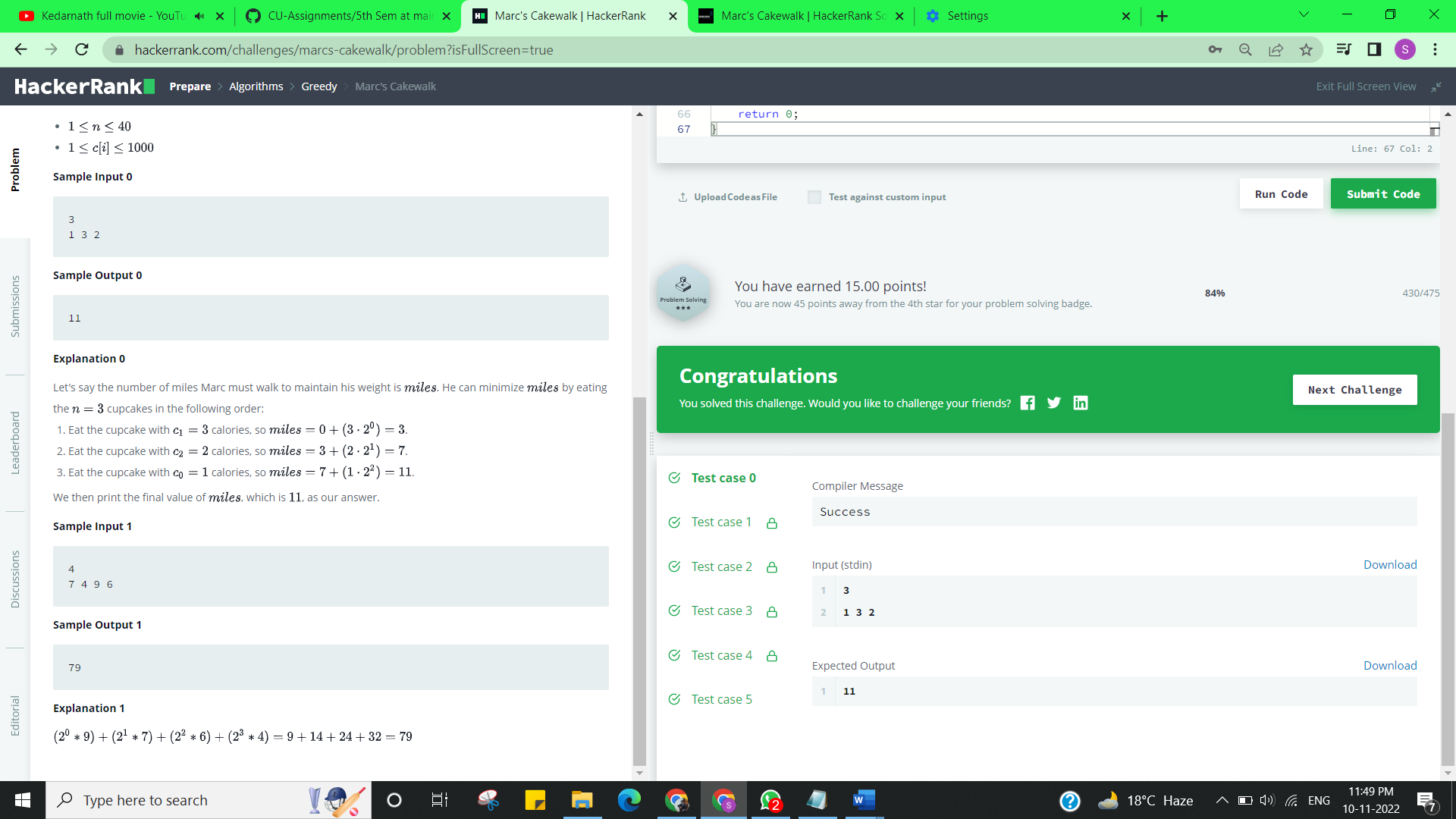
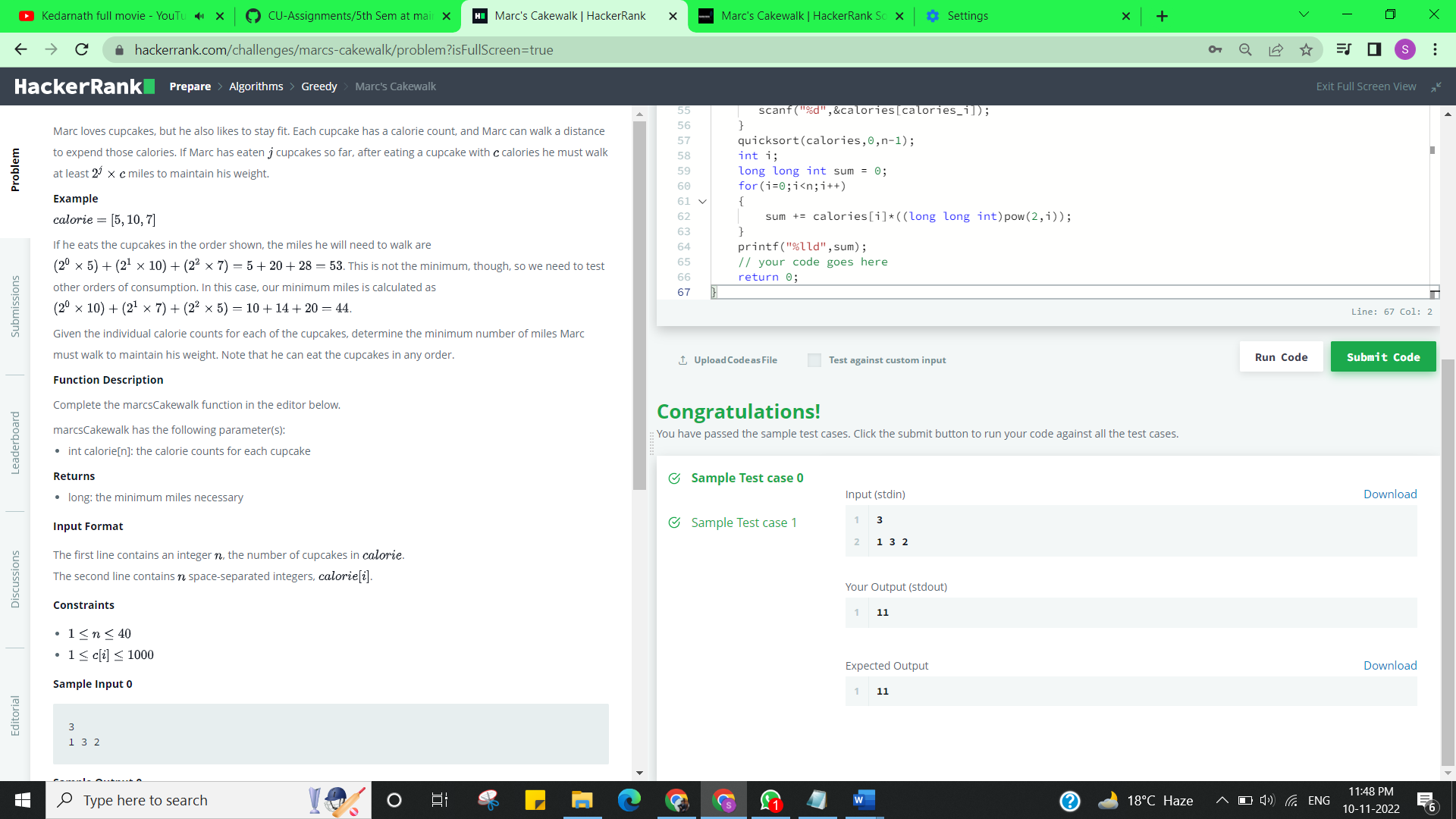
    printf("%lld",sum);

    // your code goes here

    return 0;

}

**5. Result/Output/Writing Summary:**



**Experiment 10.2**

1. **Aim/Overview of the practical:**

Branch and Bound, Greedy

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.

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

1. **Apparatus / Simulator Used:**

* Windows 7 or above
* Google Chrome

1. **Objective:**
   * To understand the concept of Branch and Bound.
   * To implement the concept of Greedy.
2. **Code:**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

typedef long long ll;

int main() {

    char m[105][105];

    ll t,n,i,j,f,cnt[26],k;

    ll a[105][105];

    scanf("%lld",&t);

    while(t--)

        {

        scanf("%lld",&n);

        for(i=0;i<n;i++)

            {

            scanf("%s",m[i]);

        }

        for(i=0;i<n;i++)

            {

            memset(cnt,0,sizeof(cnt));

            for(j=0;j<n;j++)

                {

                cnt[m[i][j]-'a']++;;

            }

            j=0;

            for(k=0;k<26;k++)

                {

                while(cnt[k]>0)

                    {

                    a[i][j]=k;

                    j++;

                    cnt[k]--;

                }

            }

        }

        f=0;

        for(j=0;j<n&&f==0;j++)

            {

            for(i=0;(i+1)<n&&f==0;i++)

                {

                if(a[i][j]>a[i+1][j])

                    {

                    f=1;

                }

            }

        }

        if(f==0)

            {

            printf("YES\n");

        }

        else

            {

            printf("NO\n");

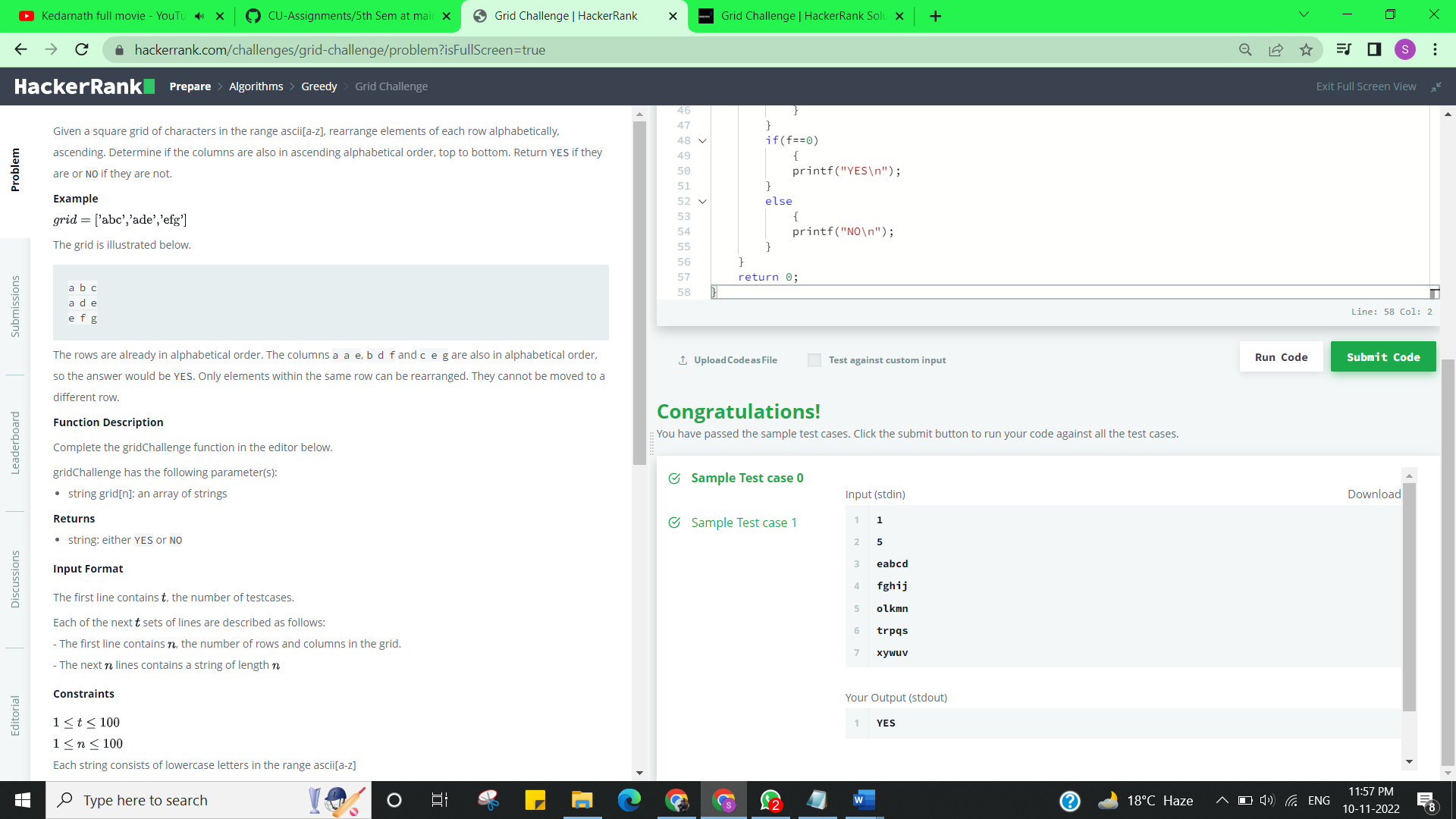
        }

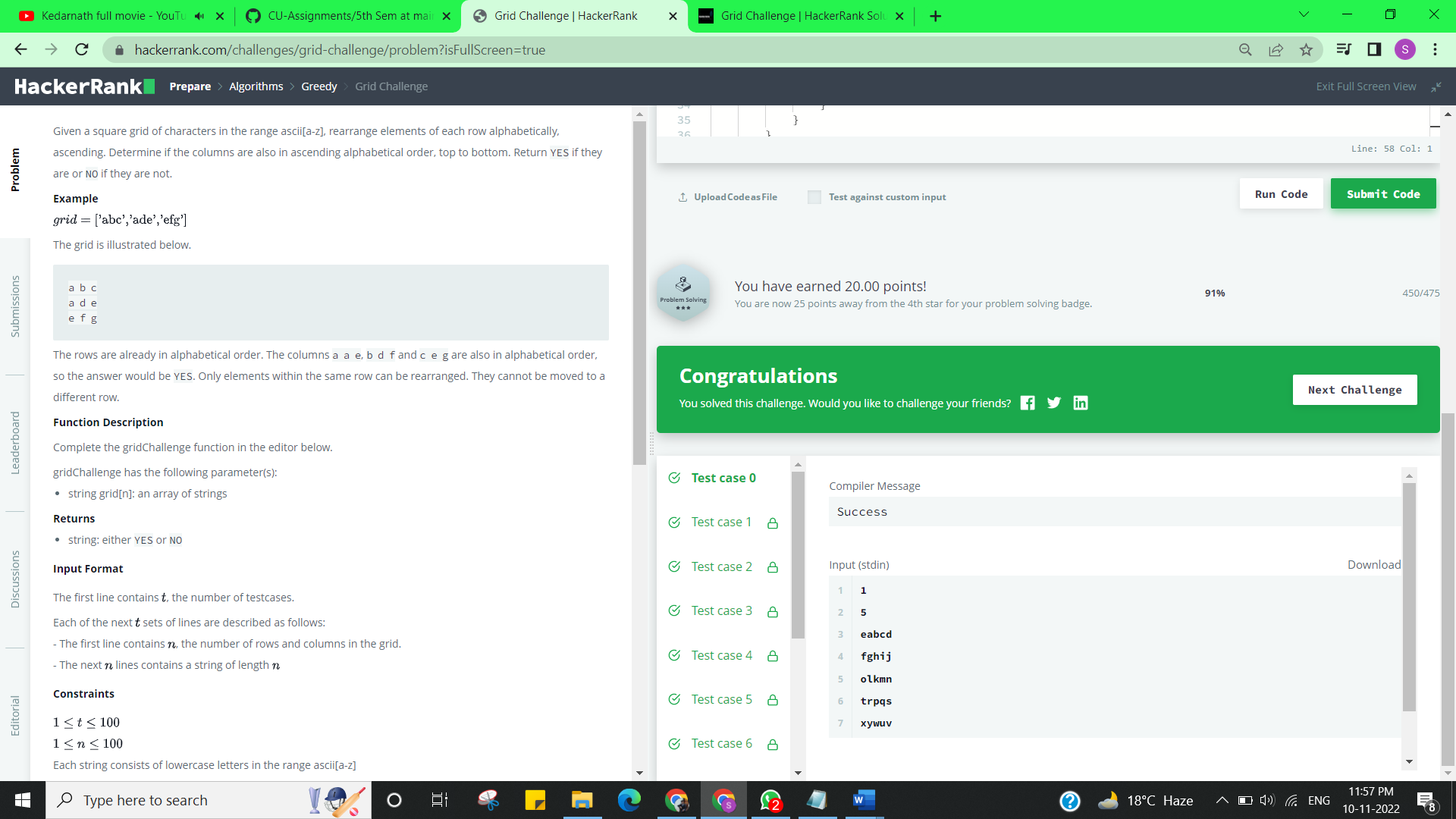
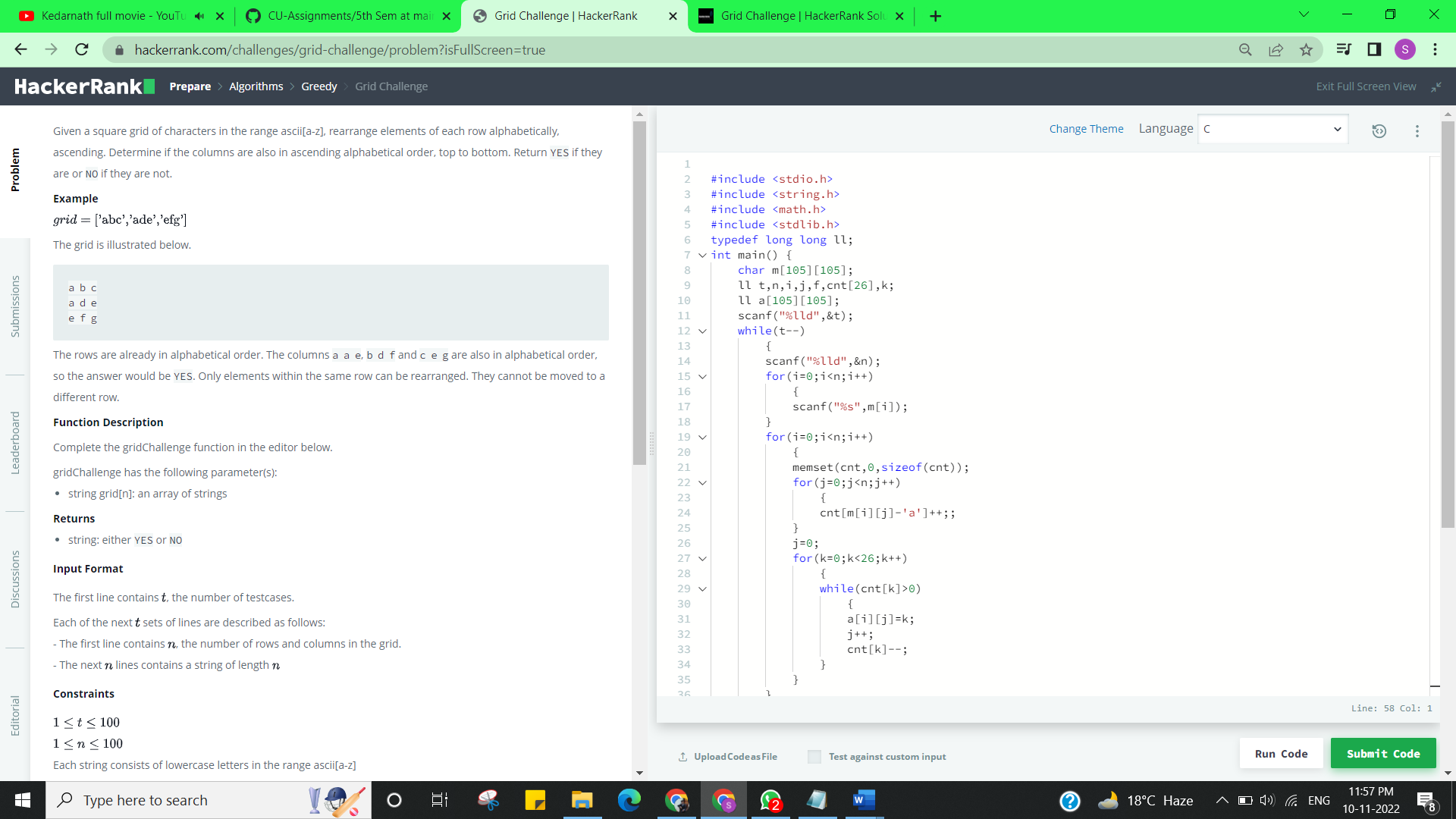
    }

    return 0;

}

1. **Result/Output/Writing Summary:**

****

****

**Learning outcomes (What I have learnt):**

* + Learned the concept of Branch and Bound.
  + Learnt about Array in Greedy.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |