**CP Club 365Days Challenge**

**Date – 01/12/2022**

**Programming language – C**

**Problem Statement**

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

**Your Code**:

// 0x45Day of 0x365Days challenge

// VEDANT BHARAD

// 1-12-2022

#include <assert.h>

#include <ctype.h>

#include <limits.h>

#include <math.h>

#include <stdbool.h>

#include <stddef.h>

#include <stdint.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char\* readline();

char\* ltrim(char\*);

char\* rtrim(char\*);

int parse\_int(char\*);

char\* myFun(char\* arr,int len)

{

    for(int loop=0;loop<len;loop++){

        for(int loop2=loop;loop2<len;loop2++){

            if(arr[loop]<arr[loop2]){

                char temp=arr[loop];

                arr[loop]=arr[loop2];

                arr[loop2]=temp;

            }

        }

    }

    return arr;

}

char\* gridChallenge(int grid\_count, char\*\* grid) {

    for(int loop=0;loop<grid\_count;loop++){

        for(int loop2=0;loop2<grid\_count;loop2++){

            for(int loop3=loop2;loop3<grid\_count;loop3++){

                if(grid[loop][loop2]>grid[loop][loop3]){

                    char temp=grid[loop][loop2];

                    grid[loop][loop2]=grid[loop][loop3];

                    grid[loop][loop3]=temp;

                }

            }

        }

    }

    for(int loop1=0; loop1<grid\_count; loop1++){

        for(int loop2=0; loop2<grid\_count-1; loop2++){

            if(grid[loop2][loop1]>grid[loop2+1][loop1]){

                return "NO";

            }

        }

    }

    return "YES";

}

int main()

{

    FILE\* fptr = fopen(getenv("OUTPUT\_PATH"), "w");

    int t = parse\_int(ltrim(rtrim(readline())));

    for (int t\_itr = 0; t\_itr < t; t\_itr++) {

        int n = parse\_int(ltrim(rtrim(readline())));

        char\*\* grid = malloc(n \* sizeof(char\*));

        for (int i = 0; i < n; i++) {

            char\* grid\_item = readline();

            \*(grid + i) = grid\_item;

        }

        char\* result = gridChallenge(n, grid);

        // fprintf(fptr, "%s\n", result);

        printf("%s\n", result);

    }

    fclose(fptr);

    return 0;

}

char\* readline() {

    size\_t alloc\_length = 1024;

    size\_t data\_length = 0;

    char\* data = malloc(alloc\_length);

    while (true) {

        char\* cursor = data + data\_length;

        char\* line = fgets(cursor, alloc\_length - data\_length, stdin);

        if (!line) {

            break;

        }

        data\_length += strlen(cursor);

        if (data\_length < alloc\_length - 1 || data[data\_length - 1] == '\n') {

            break;

        }

        alloc\_length <<= 1;

        data = realloc(data, alloc\_length);

        if (!data) {

            data = '\0';

            break;

        }

    }

    if (data[data\_length - 1] == '\n') {

        data[data\_length - 1] = '\0';

        data = realloc(data, data\_length);

        if (!data) {

            data = '\0';

        }

    } else {

        data = realloc(data, data\_length + 1);

        if (!data) {

            data = '\0';

        } else {

            data[data\_length] = '\0';

        }

    }

    return data;

}

char\* ltrim(char\* str) {

    if (!str) {

        return '\0';

    }

    if (!\*str) {

        return str;

    }

    while (\*str != '\0' && isspace(\*str)) {

        str++;

    }

    return str;

}

char\* rtrim(char\* str) {

    if (!str) {

        return '\0';

    }

    if (!\*str) {

        return str;

    }

    char\* end = str + strlen(str) - 1;

    while (end >= str && isspace(\*end)) {

        end--;

    }

    \*(end + 1) = '\0';

    return str;

}

int parse\_int(char\* str) {

    char\* endptr;

    int value = strtol(str, &endptr, 10);

    if (endptr == str || \*endptr != '\0') {

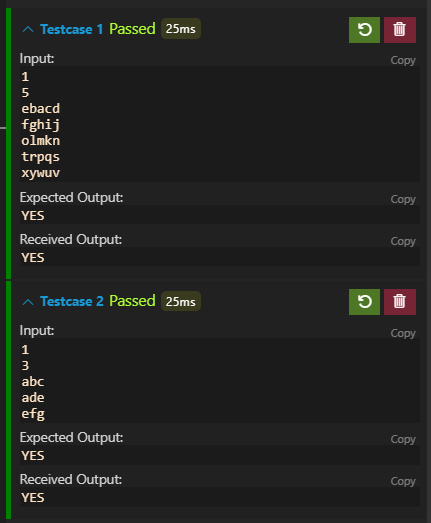
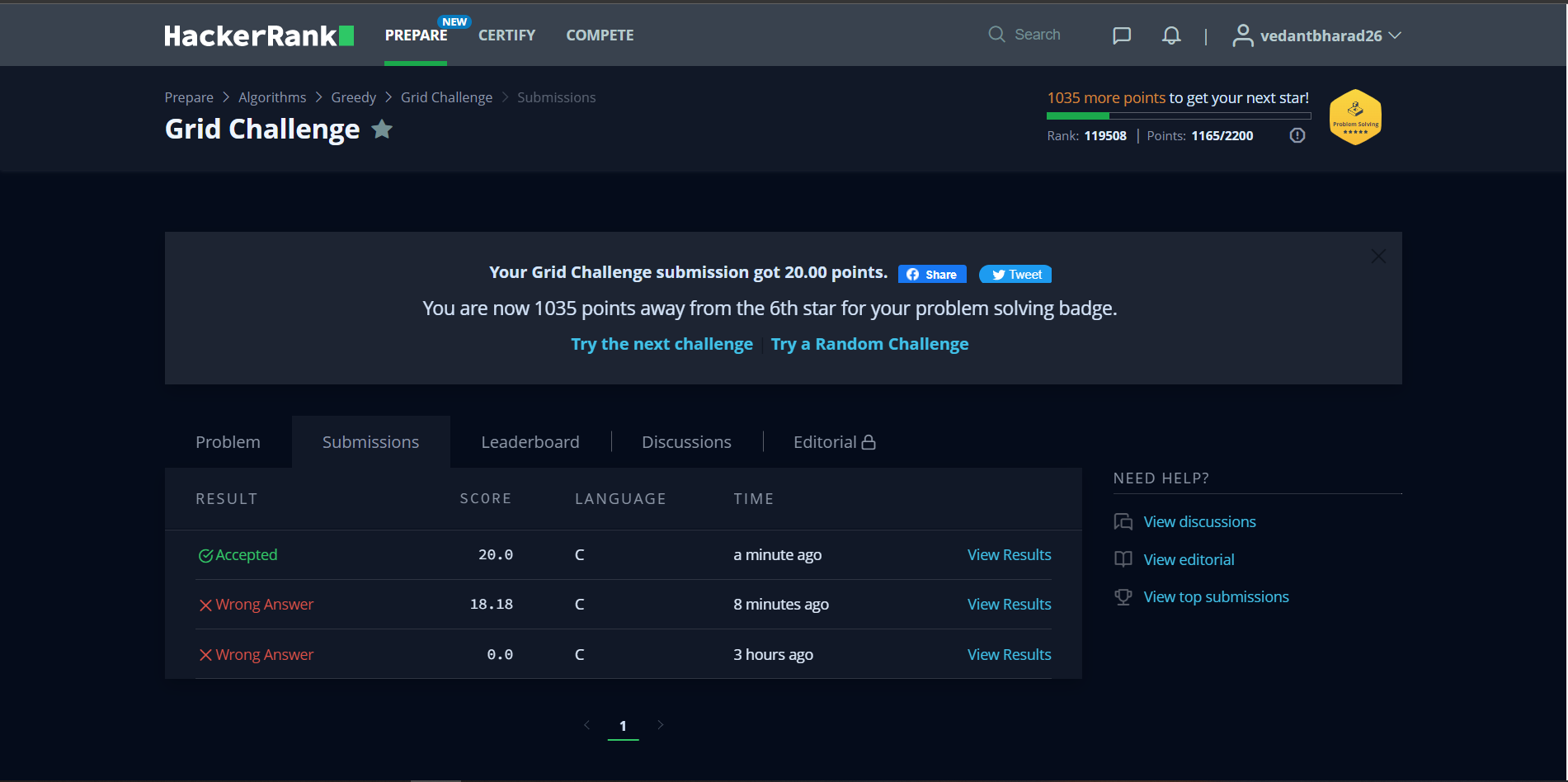
        exit(EXIT\_FAILURE);

    }

    return value;

}

**Output (Screen Shot)**:



**Understanding about problem:**

* In this task there are three inputs number of test case,length of 2D array and array it self.
* In this task I need to sort every row in ascending alphabetical order and then check columns are also in ascending alphabetical order or not, if it is in alphabetical order then return YES else NO.

Note: If you can't understand the problem, feel free to contact us and we'll help you. Please don't copy and paste from anywhere.

ALL THE BEST

Team CP Club