

## Lab Exam - 2

### Instructions

- There are 3 questions, out of which you have to attempt **all 3**.
  - For q1, there are 2 questions, out of which you have to solve **only 1**.
  - For q2, there are 2 questions, out of which you have to solve **only 1**.
  - For q3, there are 2 sub-parts, out of which you need to solve **both**.
- 

### Question 1 (30 marks)

#### Question 1.a

There are  $n$  stones on the table in a row, each of them can be red, green or blue. Count the minimum number of stones to take from the table so that any two neighboring stones had different colors. Stones in a row are considered neighboring if there are no other stones between them.

#### Input Format:

The first line contains integer  $n$  — the number of stones on the table.

The next line contains string  $s$ , which represents the colors of the stones. We'll consider the stones in the row numbered from 1 to  $n$  from left to right. Then the  $i$ -th character  $s$  equals "R", if the  $i$ -th stone is red, "G", if it's green and "B", if it's blue.

#### Output Format:

Print a single integer — the answer to the problem.

#### Constraints:

$1 \leq n \leq 50$

string  $s$  only contain characters 'R', 'G', 'B'

#### Example 1:

Input:

3  
RRG

Output:

1

Explanation: We can remove any 'R' to make the string "RG".

#### Example 2:

Input:

5

RRRRR

Output:

4

Explanation: We need to remove four 'R's to make the string "R".

**Example 3:**

Input:

4

BRBG

Output:

0

Explanation: No need to remove any character.

---

**Question 1.b**

Vasya is very upset that many people on the internet mix uppercase and lowercase letters in one word. That's why he decided to invent an extension for his favorite browser that would change the letters' register in every word so that it either only consisted of lowercase letters or, vice versa, only of uppercase ones. At that as little as possible letters should be changed in the word. For example, the word **HoUse** must be replaced with **house**, and the word **ViP** — with **VIP**. If a word contains an equal number of uppercase and lowercase letters, you should replace all the letters with lowercase ones. For example, **maTRix** should be replaced by **matrix**. Your task is to use the given method on one given word.

**Input Format:**

The first line contains a word **s** — it consists of uppercase and lowercase Latin letters and possesses the length from 1 to 100.

**Output Format:**

Print the corrected word **s**. If the given word **s** has strictly more uppercase letters, make the word written in the uppercase register, otherwise - in the lowercase one.

**Example 1:**

Input:

HoUse

Output:

house

**Example 2:**

Input:  
ViP

Output:  
VIP

**Example 3:**

Input:  
maTRIX

Output:  
matrix

**Note:** You can validate your code *here*. For that you need to register first using your institute mail id. (password: CPS)

---

**Question 2 (30 marks)****Question 2.a**

Complete the following code for implementing a circular queue using array.

```
#include <stdio.h>

int queue[10];
int front = -1, rear = -1;

int isFull() {
    // if the queue is full, return 1, else return 0
}

int isEmpty() {
    // if the queue is empty, return 1, else return 0
}

void enqueue(int q[], int value) {
    // Write your code here
}

void dequeue(int q[], int *front) {
    // Write your code here
}

void printQueue(int q[]) {
```

```

    // Write your code here
}

int main() {
    return 0;
}

```

### Question 2.b

Complete the following code for implementing a stack using array.

```

#include <stdio.h>

int stack[10];
int top = -1;

int isFull() {
    // if the stack is full, return 1, else return 0
}

int isEmpty() {
    // if the stack is empty, return 1, else return 0
}

void push(int s[], int val) {
    // Write your code here
}

void pop(int s[], int *top) {
    // Write your code here
}

void printStack(int s[]) {
    // Write your code here
}

int main() {
    return 0;
}

```

**Note:** You can also download the above boilerplate code from *here*.

---

### Question 3 (40 marks, 20 + 20)

To check your knowledge of structs, you need to solve a problem from a real-life scenario. You are given some data from the *2022 ICC Men's T20 World Cup*.

To simplify the problem, you are given the data of team **India** for the first two matches and that of only two players, namely *Rohit Sharma* and *Virat Kohli*.

Most of the code is already written for you. You need to complete the logic for the following functions:

- `fillDetails` (20 marks)
- `printAvg` (20 marks)

**Note 1:** Let us define the *average run* as the **sum of runs scored in all the matches divided by the number of matches played**. This definition differs from the actual one, but for this problem, consider the given description.

You need to solve for both to get full marks (40).

```
#include <stdio.h>
#include <string.h>

typedef struct MatchDetails {
    int runsScored;
    int ballsPlayed;
} match;

typedef struct PlayerDetails {
    char name[20];
    match matchDetails[10];
} player;

typedef struct TeamDetails {
    char name[20];
    player players[11];
} team;

void fillDetails(team *t, int playerId, char name[], int r, int b, int matchNum) {
    strcpy(t->players[playerId].name, name);
    // Complete the code below
}

void printAvg(player p[]) {
    // Write your code here
    // Sample output:
    /*
    Player Name: Rohit Sharma, Player Average: __ (float with 2 decimal points)
    Player Name: Virat Kohli, Player Average: __ (float with 2 decimal points)
    */
    for (int i = 1; i <= 2; i++) {
        float avg = 0;
```

```

        for (int j = 1; j <= 2; j++) {
            // Complete the code below
        }
    }

int main() {
    team India;
    strcpy(India.name, "India");
    fillDetails(&India, 1, "Rohit Sharma", 4, 7, 1);
    fillDetails(&India, 2, "Virat Kohli", 82, 53, 1);
    fillDetails(&India, 1, "Rohit Sharma", 53, 39, 2);
    fillDetails(&India, 2, "Virat Kohli", 62, 44, 2);
    printAvg(India.players);
}

```

**Note 2:** You can also download the above boilerplate code from [\*here\*](#).