Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year:2024), B.Sc. in CSE (Day)

LAB ASSIGNMENT NO #04

Course Title: Data Communication Lab

Course Code: CSE 308 Section: 221_D3

Lab Experiment Name: Implementing Cyclic Redundancy Check and Parity Checker

Student Details

	Name	ID
1.	Jahidul Islam	221002504

Lab Date : 09 – 03 – 2024

Submission Date : 22 – 03 – 2024

Course Teacher's Name : Sakhaouth Hossan

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… **Signature:.....**

Comments:..... **Date:.....**

1. TITLE OF THE LAB EXPERIMENT:

Implementing Cyclic Redundancy Check and Parity Checker

2. OBJECTIVES:

After complementing this lab experiment, we will gain practical knowledge and tthe outcomes of this experiment are

- To implement the parity checker.

3. PROCEDURE:

The procedure step by step:

1. **Input**:
   - The program starts by taking input, which is a binary string s.
2. **Initialization**:
   - It initializes a vector data to store characters of the input string s.
   - It initializes an integer variable **strSize** to store the size of the input string s.
3. **Copying String to Vector:**
   - The program then copies each character of the input string s into the vector data.
4. **Counting Ones:**
   - It counts the number of ones in the binary string by iterating through the vector data. This count is stored in the variable **cnt**.
5. **Manipulation Based on Count:**
   - If the count of ones **cnt** is even, it adds a '1' at the end of the binary string.
   - If the count of ones **cnt** is odd, it adds a '0' at the end of the binary string.
6. **Printing Result:**
   - Finally, it prints the modified binary string stored in the vector data.

4. IMPLEMENTATION

```
// Bismillahir Rahmanir Rahim
// jahidulZaid
#include <bits/stdc++.h>
using namespace std;
#define optimize() ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
#define endl '\n'
#define tt long long t; cin >> t;
#define ll long long
#define pb push_back

// #ifdef LOCAL
// #include "debug.h"
// #endif

// #ifdef ONLINE_JUDGE
// #include "debug.h"
// #endif

int main() {
    string s;
    cin >> s;
    int strSize = s.size();

    vector<char>data(strSize);

    for(int i = 0;i<strSize;i++){
        data[i] = s[i];
    }
    int cnt = 0;
    for (int i = 0; i < strSize; i++) {
        if (data[i] == '1') {
            cnt++;
        }
    }
    int c = strSize + 1;
    if (cnt % 2 == 0) {
/*
ths block add 1/0 at the begining of the string.
it needs to declare the vec size +1
        for (int i = c, j = c - 1; i > 0; i--, j--) {
```

```
            data[i] = data[j];
        }
*/
        // adds 1 at the end
        data.push_back('1');
    } else {
/*
ths block add 1/0 at the begining of the string.
it needs to declare the vec size +1
        for (int i = c, j = c - 1; i > 0; i--, j--) {
            data[i] = data[j];
        }
*/
        data.push_back('0');
    }
    for(auto x: data){
        cout << x;
    }
    cout<< endl;
    return 0;
}
```

…….. Continued ……

5. OUTPUT

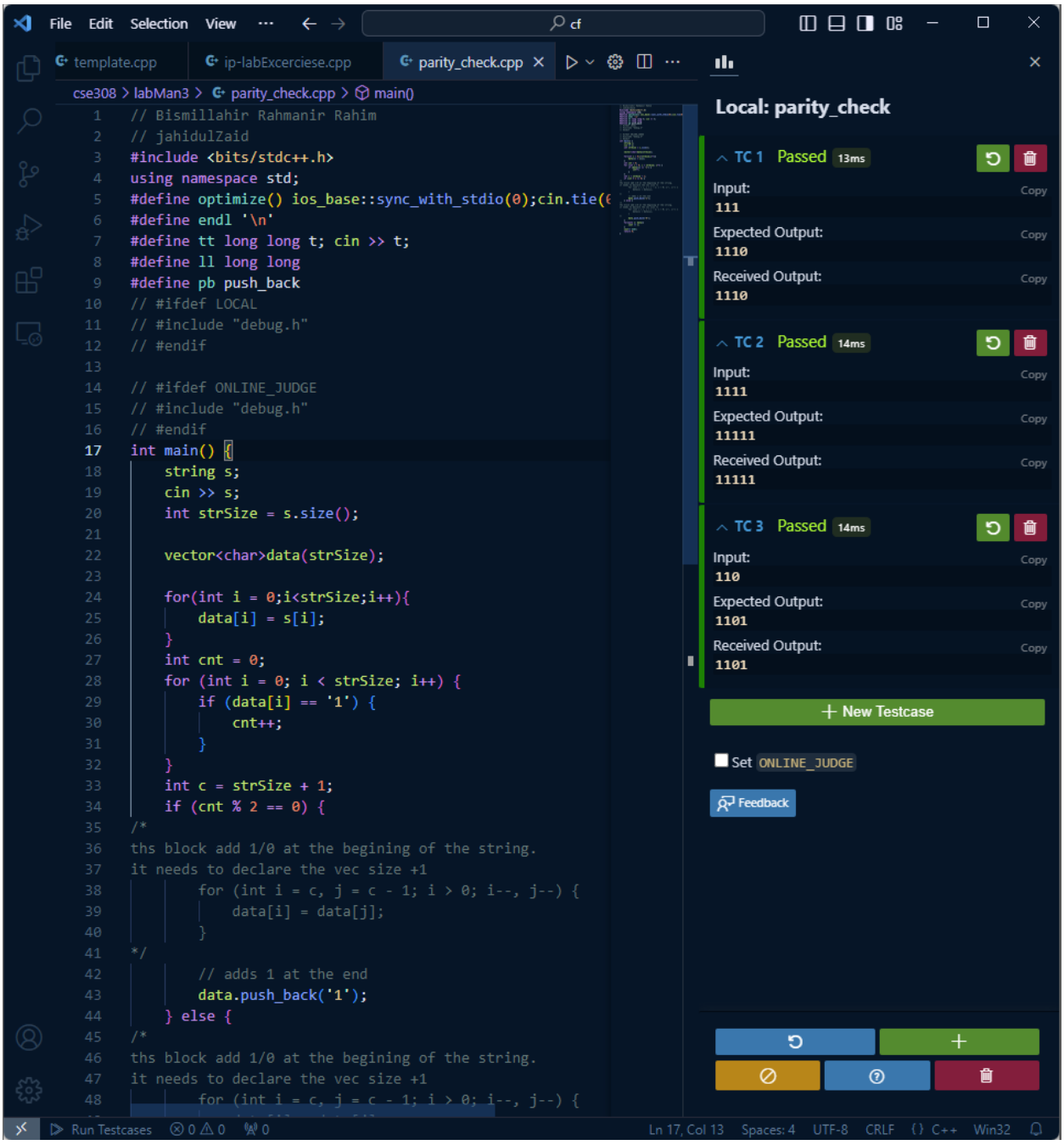

Figure 01: Shows the code and output of this code.

Figure 02: Output of the program.

6. ANALYSIS AND DISCUSSION:

- Both blocks of code add '1' or '0' at the beginning of the binary string based on whether the count of ones **cnt** is even or odd.