



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**]

<u>Lab Report Status</u>	
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 the 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

The image shows a C++ IDE with a code editor on the left and a test results panel on the right. The code in the editor is for a parity check function. It includes standard headers, defines macros for optimization and debugging, and implements a function that takes a string and returns its parity. The test results panel shows three test cases, all of which passed. The first test case has an input of '111' and an expected output of '1110'. The second test case has an input of '1111' and an expected output of '11111'. The third test case has an input of '110' and an expected output of '1101'.

```
1 // Bismillahir Rahmanir Rahim
2 // jahidulZaid
3 #include <bits/stdc++.h>
4 using namespace std;
5 #define optimize() ios_base::sync_with_stdio(0);cin.tie(0);cout.tie(0);
6 #define endl '\n'
7 #define tt long long t; cin >> t;
8 #define ll long long
9 #define pb push_back
10 // #ifdef LOCAL
11 // #include "debug.h"
12 // #endif
13
14 // #ifdef ONLINE_JUDGE
15 // #include "debug.h"
16 // #endif
17 int main() {
18     string s;
19     cin >> s;
20     int strSize = s.size();
21
22     vector<char>data(strSize);
23
24     for(int i = 0; i < strSize; i++){
25         data[i] = s[i];
26     }
27     int cnt = 0;
28     for (int i = 0; i < strSize; i++) {
29         if (data[i] == '1') {
30             cnt++;
31         }
32     }
33     int c = strSize + 1;
34     if (cnt % 2 == 0) {
35         /*
36         ths block add 1/0 at the beginning of the string.
37         it needs to declare the vec size +1
38         for (int i = c, j = c - 1; i > 0; i--, j--) {
39             data[i] = data[j];
40         }
41         */
42         // adds 1 at the end
43         data.push_back('1');
44     } else {
45         /*
46         ths block add 1/0 at the beginning of the string.
47         it needs to declare the vec size +1
48         for (int i = c, j = c - 1; i > 0; i--, j--) {
```

Local: parity_check

TC 1 Passed 13ms

Input: 111
Expected Output: 1110
Received Output: 1110

TC 2 Passed 14ms

Input: 1111
Expected Output: 11111
Received Output: 11111

TC 3 Passed 14ms

Input: 110
Expected Output: 1101
Received Output: 1101

+ New Testcase

Set ONLINE_JUDGE

Feedback

Run Testcases 0 0 0 0

Ln 17, Col 13 Spaces: 4 UTF-8 CRLF () C++ Win32

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.

