Assignment 8

ELP -718 Telecommunication Software Laboratory

Priyanka Singh

2019JTM2087

2019-2021

A report presented for the assignment on

Python

Bharti School Of

Telecommunication Technology and Management

IIT Delhi

India

September 18, 2019

Contents

List of Figures

1 Problem Statement-1

1.1 Problem Statement

Parity Check The simplest way of error detection is to append a single bit, called a parity check, to a string of data bits. This parity check bit has the value 1 if the number of 1âĂŹs in the bit string is even and has the value 0 otherwise, i.e., Odd Parity Check.

Bit-Oriented Framing Data Link Layer needs to pack bits into frames so that each frame is distinguishable from another. Frames can be fixed or variable size. In variable size framing, we define the end of the frame using a bit-oriented approach. It uses a special string of bits, called a flag for both idle fills and to indicate the beginning and the ending of frames. The bit stuffing rule is to insert a 0 after each appearance of 010 in the original data. The string 0101 is used as the bit string or flag to indicate the end of the frame.

1.2 Assumptions

1. Using linux environment through Ubuntu.
2. Using Latex Libraries and Packages.
3. Using Eclipse Ide,gcc and gnu make for c code compilation and execution.

1.3 Program Structure

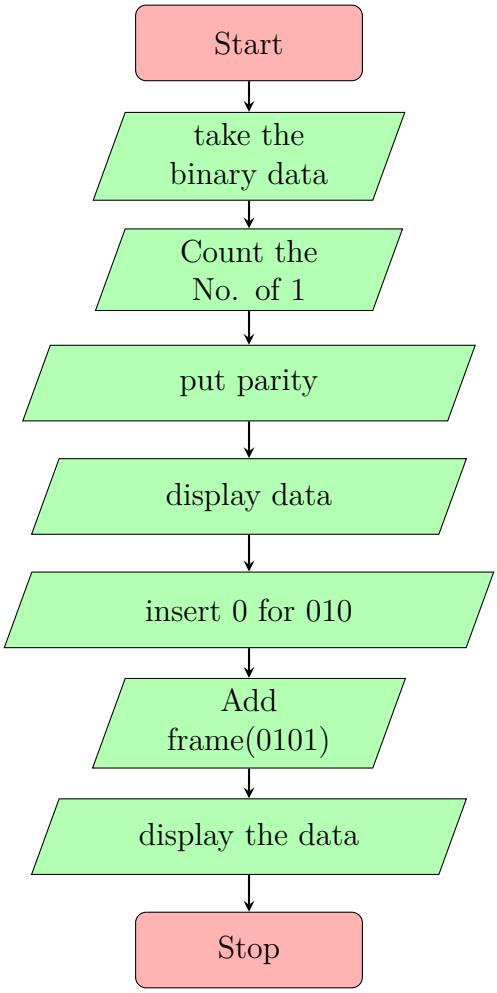

Figure 1: Problem 1 Flowchart

1.4 Algorithm and Implementation

1. start
2. Enter binary string by user.
3. Count no. of 1 's in the string.
4. If no of 1's is even , append a 1 at the end of the binary string otherwise append 0.
5. Find a substring 010 in the string.
6. if end of frame replace it by 0100.
7. Print parity corrected string and transmitted bite
8. exit

1.5 Input-Output format

Input

Enter binary bit data that has to be transmitted.

Output

1.6 Test case

1.7 Difficulty Faced

- Error code compiling.
- Adding proper inputs.
- Adding proper comments.
-

1.8 Screenshots

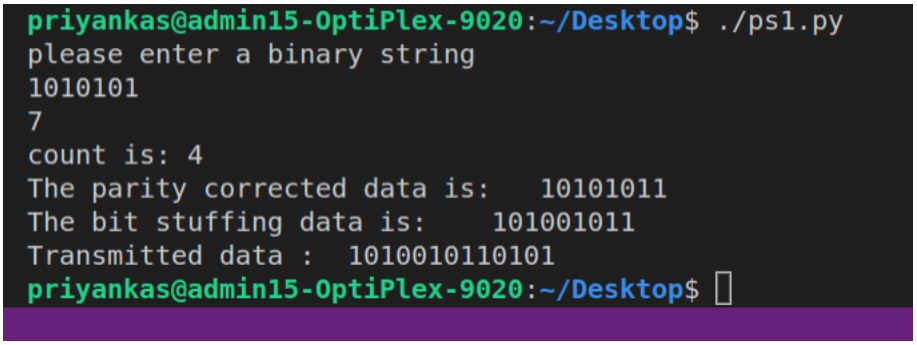

Figure 2: Screenshot1

2 Problem Statement-2

2.1 Problem Statement

. 3X3 Numeric Tic-Tac-Toe (Use numbers 1 to 9 instead of XâĂŹs and OâĂŹs) One player plays with the odd numbers (1, 3, 5, 7, 9) and the other player plays with the even numbers (2,4,6,8). All numbers can be used only once. The player who puts down 15 points in a line wins (sum of 3 numbers). Always Player with odd numbers starts the game. Once a line contains two numbers whose sum is 15 or greater, there is no way to complete that line, although filling in the remaining cells might be necessary to complete a different line. Note âĂŞ Line can be horizontal, vertical or diagonal **terminal:**

- Print 'Welcome to the Game '
- Print whether it is Player 1 's or Player 2 's chance.
- Get the position and number to be entered from the user.
- Show tic tac toe with data.
- Continue till the game gets draw or some player wins and show the result.
- Ask the user whether to continue for the next game or exit.

  s

2.2 Assumptions

1. Using linux environment through Ubuntu.
2. Using Latex Libraries and Packages.
3. Using Eclipse Ide,gcc and gnu make for c code compilation and execution.

2.3 Program Structure

2.4 Algorithm and Implementation

1. start
2. display the game board
3. using list named occpos for showing occupied positions
4. take value of no. and the position to insert from the user
5. check for the horizontal, vertical, and diagonal win
6. continue till the winner is declared
7. stop

1. End

2.5 Input-Output format

Input I

Output

2.6 Difficulty Faced

- Error code compiling.
- Adding proper inputs.
- Adding proper comments.

2.7 Screenshots

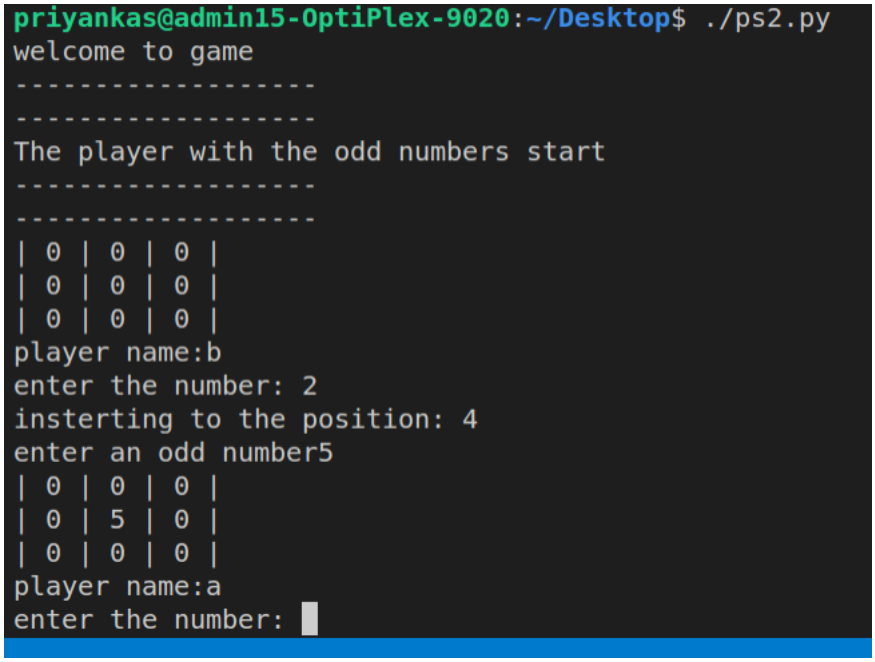

Figure 3: Screenshot1

Figure 4: Screenshot1

3 Appendix

3.1 Code for ps1

3.2 Code for ps2

References

[1] BitBucket. *Learn Git*. https://www.atlassian.com/git/tutorials.

[2] GeeksForGeeks. *List methods in Python*. https://www.geeksforgeeks.org/list-methods-python/.

[3] Python Software Foundation. *Python 3.7.4 documentation*. https://docs.python.org/3/.

[4] Telusko. *Python Programming Tutorial for Beginners*. https://www.youtube.com/watch?v=QXeEoD0pB3E&list=PLsyeobzWxl7poL9JTVyndKe62ieoN-MZ3.

[5] w3schools. *Python Dictionaries*. https://www.w3schools.com/python/python_dictionaries.asp.