Assignment-8

ELL - 718 Telecom Software Laboratory

Kala Kanhu Karsi

2019JTM2836
2019-2020

A report presented for the assignment on

Python Basics & Github

Bharti School Of

Telecommunication Technology and Management (EE)

IIT Delhi

India

September 18, 2019

Contents

List of Figures

1 Problem Statement-1

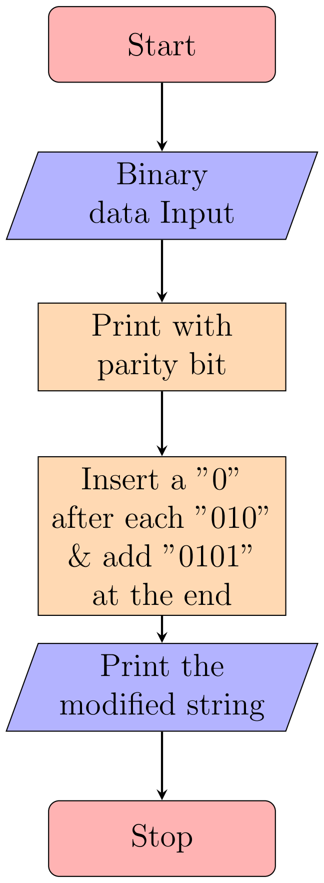
1.1 Problem Statement

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 1s 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

- The output is shown by taking the screeenshots of terminal.
- The codes are written using "gedit editor"

1.3 Program Structure


Start
Binary data Input
Print with parity bit
Insert a ”0” after each ”010” & add ”0101” at the end
Print the modified string
Stop

1.4 Algorithm and Implementation

- Input binary bit data that has to be transmitted.
- Print binary bit data with parity bit i.e. Odd parity bit.
- 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.
- Print the modified string that is to be transmitted.

1.5 Input and Output Format

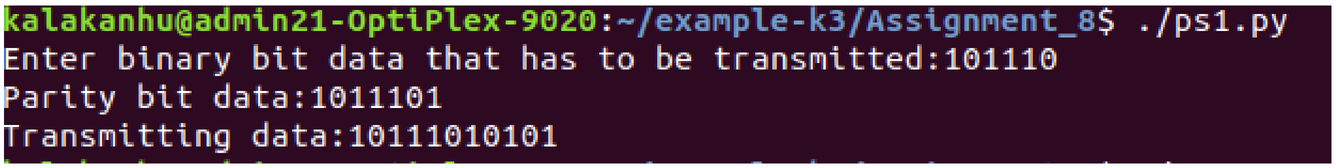
Input Format
Enter binary bit data that has to be transmitted.
Output Format
Print binary bit data with parity bit.
Print the modified string that is to be transmitted.

1.6 Difficulties/Issues Faced

- Difficulty faced while inserting a 0 after each appearance of 010 in the original data.

1.7 Screenshots

Figure 1: screenshot1



```
kalakanhu@admin21-OptiPlex-9020:~/example-k3/Assignment_8$ ./ps1.py
Enter binary bit data that has to be transmitted:101110
Parity bit data:1011101
Transmitting data:10111010101
```

2 Problem Statement-2

2.1 Problem Statement

3X3 Numeric Tic-Tac-Toe (Use numbers 1 to 9 instead of Xs and Os)
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 Line can be horizontal, vertical or diagonal
Constraints:

- 1<=Position<=9
- 1<=Number<=9

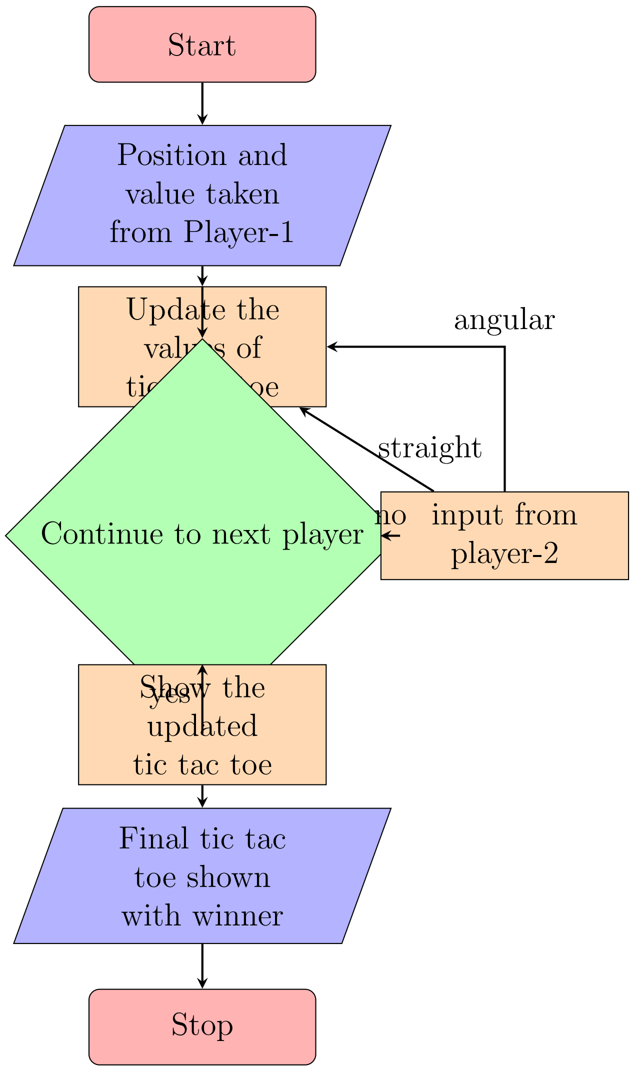
Terminal:

- Print Welcome to the Game!.
- Print whether it is Player 1s or Player 2s 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.

2.2 Assumptions

- The output is shown by taking the screeenshots of terminal.
- The codes are written using "gedit editor"

2.3 Program Structure


Start
Position and value taken from Player-1
Update the values of tic tac toe
angular
straight
Continue to next player
no
input from player-2
yes
Show the updated tic tac toe
Final tic tac toe shown with winner
Stop

2.4 Algorithm and Implementation

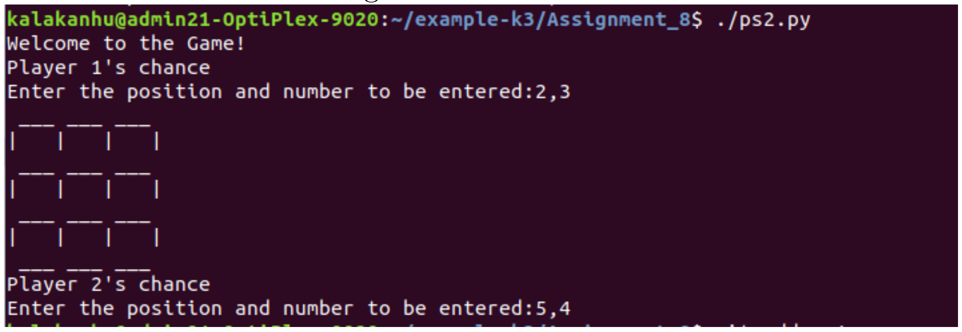
- First the player who plays with odd numbers was given chance to enter position and value.
- Tic tac toe was shown with the data fed by player-1
- Now player-2 was given chance to enter position and value.
- Tic tac toe was shown with the data fed by player-1
- This continues till the game gets draw or some player wins and shown the result.
- The user was asked whether to continue for the next game or exit.

2.5 Difficulties/Issues Faced

- Difficulty faced while making the tic-tac-toe.
- Difficulty faced while entering the position and value.

2.6 Screenshots

Figure 2: screenshot2



```
kalakanhu@admin21-OptiPlex-9020:~/example-k3/Assignment_8$ ./ps2.py
Welcome to the Game!
Player 1's chance
Enter the position and number to be entered:2,3
 ___ ___ ___
|   |   |   |
 ___ ___ ___
|   |   |   |
 ___ ___ ___
|   |   |   |
 ___ ___ ___
Player 2's chance
Enter the position and number to be entered:5,4
```

3 Appendix

3.1 Appendix-A : Code for ps1

```
#!/usr/bin/python3
#parity-check
#input binary message
input_bin =str(input ("Enter binary bit data that has to be transmitted:"))
#Defining a function to count no. of occurrences of 1
count = 0
for i in input_bin:
if i == '1':
count = count + 1
#taking binary digits as input
#count checking if even or odd
#n=len(bin_list)
if (count%2==0):
input_bin = input_bin + "1"
#bin_list = bin_list.split("")
print ("Parity bit data:{0}".format(input_bin))
else:
input_bin = input_bin + "0"
#bin_list = bin_list.split("")
print ("Parity bit data:{0}".format(input_bin))

#Bit-oriented framing
string = input_bin

# Prints the string by replacing only 3 occurence
str2 = string.replace("010", "0100", 3)
#appending 0101 at end
str1=str2+"0101"
print("Transmitting data:{0}".format(str1))
```

3.2 Appendix-B : Code for ps2

```
#!/usr/bin/python3
print("Welcome to the Game!")
def drawsmall():
a = (' ___' *  3 )
b = '   '.join('||||')
print('\n'.join((a, b, a, b, a, b, a, )))

drawsmall()
list1 =[0,0,0,
0,0,0,
0,0,0]
print(list1)
print("Player 1's chance")
pos,num=input("Enter the position and number to be entered:").split(",")

print("Player 2's chance")
pos,num=input("Enter the position and number to be entered:").split(",")
```

References

[1] Atlassian Bitbucket. *Git tutorials and Training — Atlassian Git Tutorials.* https://www.atlassian.com/git/tutorials.

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

[3] Geekforgeeks. *Python Tutorial.* https://www.geeksforgeeks.org/python-programming-examples/.

[4] MicrowaveSam. *How to Get Started with Github - Beginner Tutorial.* https://www.youtube.com/watch?v=73I5dRucCds.

[5] Tutorials Point. *Python Tutorial.* https://www.tutorialspoint.com/python/index.htm.

[6] w3schools.com_THE WORLD'S LARGEST WEB DEVELOPER SITE. *Python Programming Examples.* https://www.w3schools.com/python/.