Assignment-4

ELP - 718 Telecom Software Laboratory

Saubhadra Gautam

2019JTM2167

2019-21

A report presented for the assignment on
Telecommunication software lab

Bharti School of
Telecommunication Technology and Management
IIT DELHI
India

July 27,2019

Contents

List of Figures

1 Problem statement 1

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 1's 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.

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

Sample Input

0101011101001 01

Sample Output

Parity bit data : 0101011101001011
Transmitting data: 01001011101000100110101

1.1 Objective

You have to create a bit and check its parity bit and add 0 if odd parity occur .
You have to see for sequence and append 0 if it occur

1.2 Screenshots

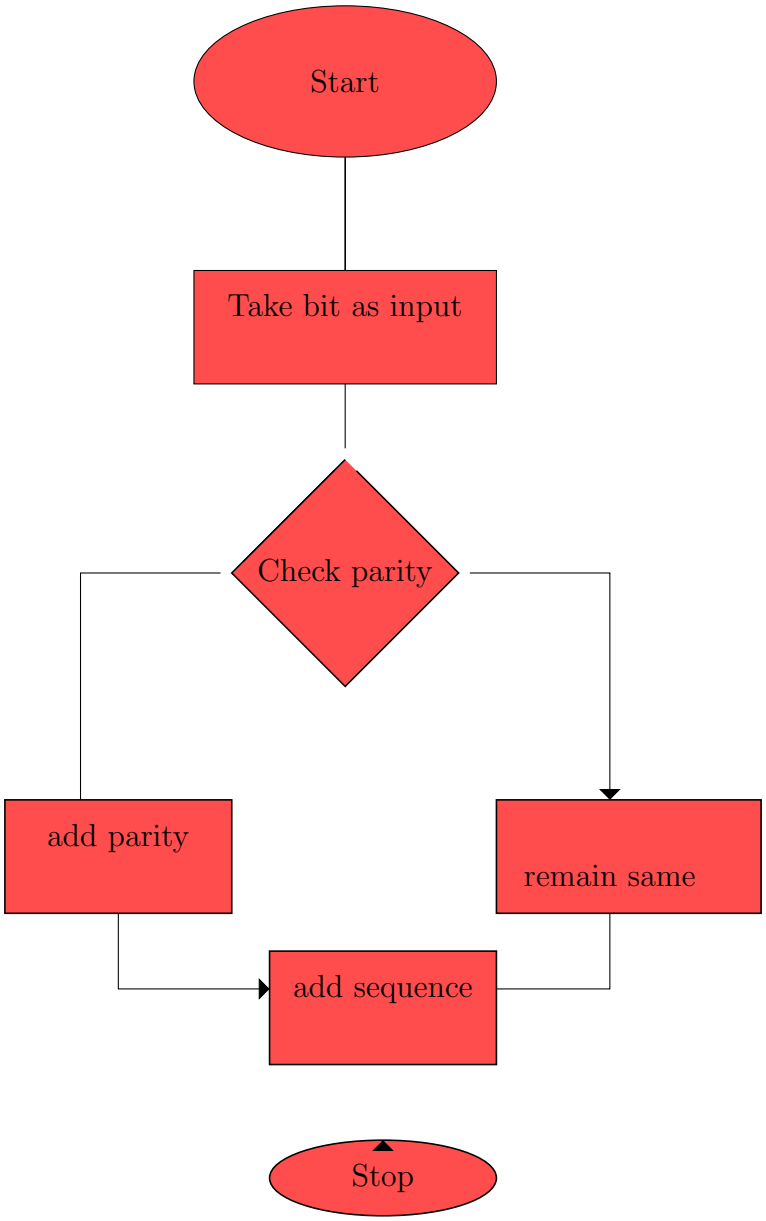
```
/usr/bin/python3.6 /home/saubhadra/2019jtm2167_8/Assignment_8/ps1.py
Enter the transmitted bit :1010101
[1, 0, 1, 0, 1, 0, 1]
Parity bit data
10101011
[1, 0, 1, 0, '0', 1, 0, 1, 1]
101001011
Transmitting data:
1010010110101
```

Figure 1

```
/usr/bin/python3.6 /home/saubhadra/2019jtm2167_8/Assignment_8/ps1.py
Enter the transmitted bit :1010101
[1, 0, 1, 0, 1, 0, 1]
Parity bit data
10101011
[1, 0, 1, 0, 1, 1, 0, 1, 1]
Transmitting data:
101011011

Process finished with exit code 0
```

Figure 2: Transmitted data

1.3 Flowchat


Start
Take bit as input
Check parity
add parity
remain same
add sequence
Stop

1.4 Assumptions

- Proper Pycharm or other editor and error resolving skills.
- Should able to build the logic of program.
- Several variable are taken at the time of code which are used in the program
- hand on experience on Pycharm IDE and Python language.
- Proper Knowledge of Python programming and its attributes.
- Knowledge of git and git hub.

1.5 Difficulties /Issues faced

- Forming logic for the code and implementation.
- Not able to do the logic building.
- several encounter of segmentation error and code dumped error.
- Problem faced on Pycharm and its code runner extension.
- Formatting and indentation in the code.

1.6 Algorithm

- Take input bit stream from user at runtime.
- Typecast it to integer.
- Split bit stream in list of numbers.
- Check odd parity.
- If yes then append 0 as parity bit at last else take it same.
- Find 010 pattern and insert 0 if it occur in bit sequence.
- Transmit bit by adding 0101 at last.
- Print resultant transmitted bit on console.

2 Problem Statement 2

3X3 Numeric Tic-Tac-Toe (Use numbers 1 to 9 instead of X's and O's)

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:

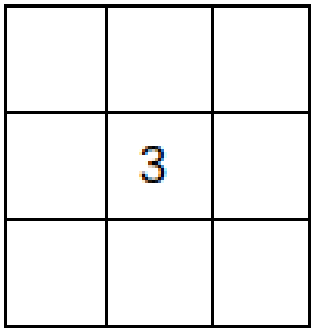
- Print 'Welcome to the Game!'.
- Print whether it is Player 1's or Player 2's chance.
- Get the position and number to be entered from the user.
- Show tic tac toe with data.
- Continue till the game gets draw or some player wins and show the result.
- Ask the user whether to continue for the next game or exit.
- $1 <= Number <= 9$

Output Format

Welcome to the Game!

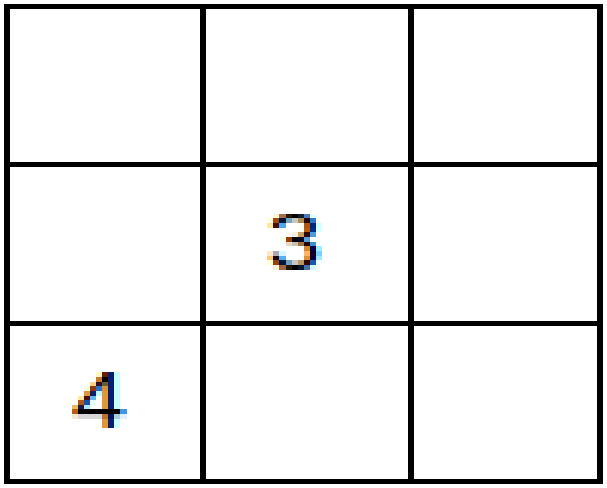
Player 1's chance

Enter the position and number to be entered: 5,3



Player 2's chance

Enter the position and number to be entered: 7,4



Continue till game ends

Note – Must use at least one User Defined Function.

2.1 Objective

We have to form a 3X3 Numeric Tic-Tac-Toe and print the winner name at last of the every game.
It should be user defined which player to play first and we have to declare draw if no result came at end

2.2 Screenshots

```
Enter 1 : game
 2 : exit1
Welcome to the Game!
Enter player 1 or 2 :1
Player1 turn
player 1 can add [1,3,5,7,9] numbers only
player 2 can add [2,4,6,8] numbers only
Enter 1 to exit else for play2
Enter the position :2
Enter the number :3
[0, 3, 0]
[0, 0, 0]
[0, 0, 0]
3
Player2 turn
```

Figure 3: second code

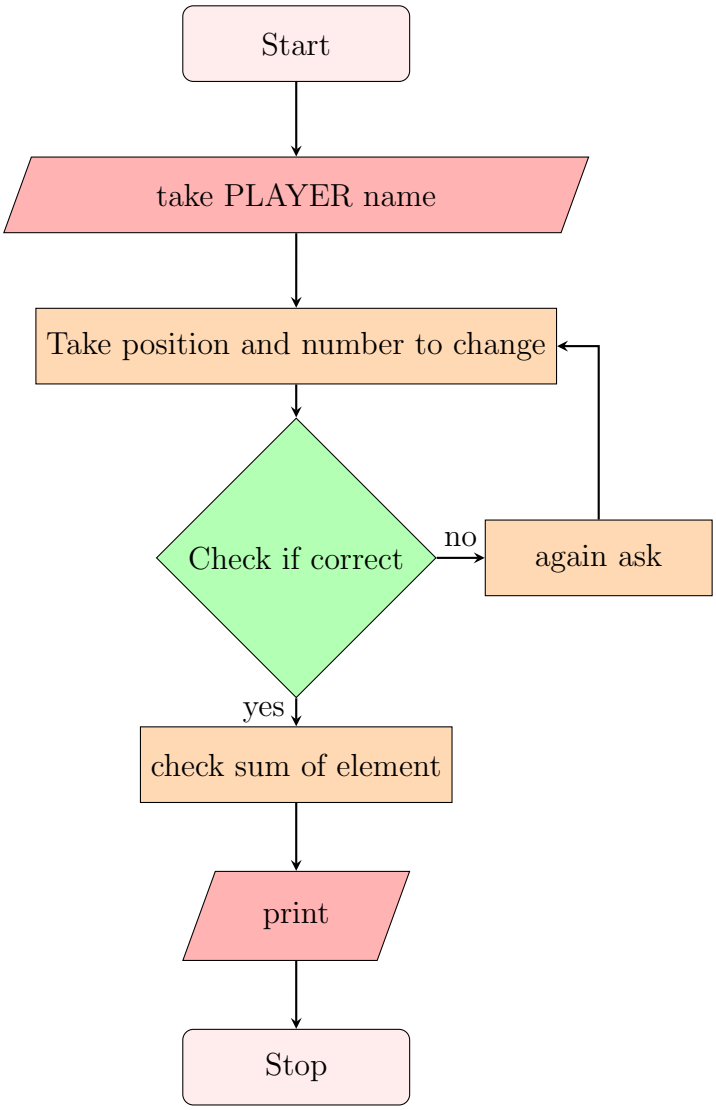
```
/usr/bin/python3.6 /home/saubhadra/2019jtm2167_8/Assignment_8/ps2.py
Welcome to the Game!
Enter player 1 or 2 :1
Player1 turn
Enter the position :3
Enter the number :3
[0, 0, 3]
[0, 0, 0]
[0, 0, 0]
3
Player2 turn
Enter the position :2
Enter the number :8
[0, 8, 3]
[0, 0, 0]
[0, 0, 0]
11
Player1 turn
Enter the position :5
Enter the number :5
[0, 8, 3]
[0, 5, 0]
[0, 0, 0]
11
Player2 turn
Enter the position :1
Enter the number :4
[4, 8, 3]
[0, 5, 0]
[0, 0, 0]
15
Player1 turn
Element Exists and you win
Player2 wins

Process finished with exit code 0
```

Figure 4: second code2

```
Welcome to the Game!
Enter player 1 or 2 :1
Player1 turn
palyer 1 can add [1,3,5,7,9] numbers only
palyer 2 can add [2,4,6,8] numbers only
Enter the position :2
Enter the number :6
[0, 6, 0]
[0, 0, 0]
[0, 0, 0]
6
Player2 turn
Enter the position :
```

Figure 5: second code3

2.3 Flowchat



2.4 Assumptions

- Proper Pycharm or other editor and error resolving skills.
- Should able to build the logic of program.
- Several variable are taken at the time of code which are used in the program
- hand on experience on Pycharm IDE and Python language.
- Proper Knowledge of Python programming and its attributes.
- Knowledge of git and git hub.

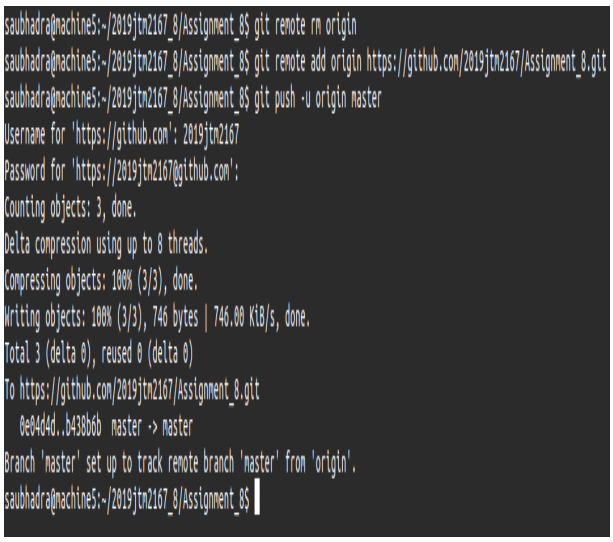
2.5 Difficulties /Issues faced

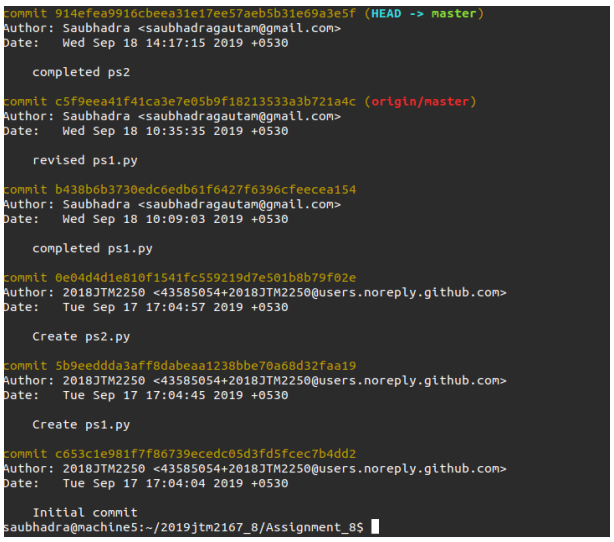
- Forming logic for the code and implementation.
- Not able to do the logic building.
- several encounter of segmentation error and code dumped error.
- Problem faced on Pycharm and its code runner extension.
- Formatting and indentation in the code.

2.6 Algorithm

- Take input as two list with even and odd numbers
- Ask for player and assign it one list other list to another player.
- Ask location and number to add.
- Append the number in the location in matrix.
- Sum each row and coloum iteratively
- Check for sum - 15 of any row or coloum
- If sum = 15 then declare that player as winner
- Print resultant transmitted bit on console.

3 git repository





4 Appendix

$https://github.com/2019jtm2167/Assignment_8$

4.1 Code for Problem 1

```
1
2
3  def concatenate_list_data(list):
4      result= ''
5      for element in list:
6          result += str(element)
7      return result
8  #
9  def Reverse(lst):
10     lst.reverse()
11     return lst
12
13 # List initialization   # Converting to int
14 bit=int(input("Enter the transmitted bit :"))
15 count1=0
16 count0=0
17
18 # Output list initialization
19 output = []
20
21 while (bit > 0):
22     rem = bit % 10
23     bit = int(bit / 10)
24     output.append(rem)
25 print(Reverse(output))
26 # Printing output
27 # print(output)
28 # print(output[2])
29 i=0
30 for i in range(0, len(output)):          #  finding  remender
31     if(output[i]==1):
32         count1+=1
33
34 remen = count1 / 10
35
36 if(count1%2==0):
37     # print("This Number is Even")          # finding even or odd
38     output.append(1)
39     # print(output)
40 else:
41     print(output)
42 ########################################## second code
      ################################
43 print("Parity bit data")
44 print(concatenate_list_data(output))              # calling function for
      concatenate
45 buffer=concatenate_list_data(output)
46 for i in range(0, len(output)):
47     if (output[i] == 0 and output[i+1] == 1 and output[i+2] == 0):         #
      checking for sequence
48         output.insert(i+3,"0")
```

```
49 print(output)
50
51 print(concatenate_list_data(output))
52 buffer=concatenate_list_data(output)+"0101"
53 print("Transmitting data:")
54 print(buffer)
```

4.2 Code for Problem 2

```
def Cloning(li1):
    li_copy = li1[:]
    return li_copy


# Function to calculate sum of each row
def row_sum(matrix):
    sum1 = 0
    sum2 = 0
    sum3 = 0

    # print("\nFinding Sum of each row:\n")

    # finding the row sum

    sum1 = matrix[0] + matrix[1] + matrix[2]
    sum2 = matrix[3] + matrix[4] + matrix[5]              # adding row elemnt
    sum3 = matrix[6] + matrix[7] + matrix[8]

            # Print the row sum
    # print("Sum of the row 1 =", sum1)
    # print("Sum of the row 2 =", sum2)
    # print("Sum of the row 3 =", sum3)

        # Reset the sum
    # sum = 0


# Function to calculate sum of each column
def column_sum(matrix):
    sum4 = 0
    sum5 = 0
    sum6 = 0

    # print("\nFinding Sum of each column:\n")
    # print("\nFinding Sum of each diagonal:\n")

    # finding the row sum

    sum4 = matrix[0] + matrix[3] + matrix[6]
    sum5 = matrix[1] + matrix[4] + matrix[7]          # adding coloumn element
    sum6 = matrix[2] + matrix[7] + matrix[8]

            # Print the row sum
    # print("Sum of the column 1 =", sum4)
    # print("Sum of the column 2 =", sum5)
    # print("Sum of the column 3 =", sum6)
while(1):
    # label1: check
    game=int(input("Enter 1 : game \n 2 : exit"))
    if(game==1):
        count=0
        print (u'Welcome to the Game!')
        # Initialize matrix
        matrix = [0,0,0,0,0,0,0,0,0]
        list1 = [1,3,5,7,9]
        list2 = [2,4,6,8]
```

```
58          play=int(input("Enter player 1 or 2 :"))
59          if(play==1):
60              play1=Cloning(list1)
61              play2=Cloning(list2)        # cloning the list to player
62              print("Player1 turn")
63              print("player 1 can add [1,3,5,7,9] numbers only")
64              print("player 2 can add [2,4,6,8] numbers only")
65          elif(play==2):
66              play1=Cloning(list2)
67              play2=Cloning(list1)            # cloning the list to player
68              print("player2 turn")
69              print("player 2 can add [1,3,5,7,9] numbers only")
70              print("player 1 can add [2,4,6,8] numbers only")
71          else:
72              print("wrong choice")
73          while(21):
74              #int(play1)
75              cede=int(input("Enter 1 to exit else for play"))
76              if(cede==1):
77                  print("Bye")
78                  exit()
79
80              pos=int(input("Enter the position :"))        # Enter the location
81              num=int(input("Enter the number :"))
82              # if(play ==
83              # for i in range(3):     # A for loop for row entries  # A for loop
    for column entries
84              #     a =[]
85              #     for j in range(3):
86              #         z=i+j
87              #         if (z == pos):
88              #             print("hello1")
89              #             a.insert(num)
90              #     matrix.append(a)
91              # print("hello")
92              # print(matrix)
93              # For printing the matrix
94              # for i in range(3):
95              #   for j in range(3):
96              #     print(matrix[i][j], end = ' ')
97              #   print()
98              count+=1
99              for i in range(9):
100               if (i==pos):
101                 matrix[i-1]=num
102
103             # print( matrix)
104             print(matrix[0:3])
105             # print("\n")
106             print(matrix[3:6])
107             # print("\n")
108             print(matrix[6:9])
109             # print("\n")
110
111
112             # for i in range(3):
113             #   for j in range(3):
114             #     print(matrix[i][j], end = ' ')
115             #   print()
116             # Get each row sum
```

```
            row_sum(matrix)

            # Get each column sum
            column_sum(matrix)
            sum1 = matrix[0] + matrix[1] + matrix[2]
            sum2 = matrix[3] + matrix[4] + matrix[5]
            sum3 = matrix[6] + matrix[7] + matrix[8]
            sum4 = matrix[0] + matrix[3] + matrix[6]
            sum5 = matrix[1] + matrix[4] + matrix[7]
            sum6 = matrix[2] + matrix[7] + matrix[8]
            sum7 = matrix[0] + matrix[4] + matrix[8]
            sum8 = matrix[2] + matrix[4] + matrix[6]
            # print(sum8)
            list1 = [sum1, sum2, sum3, sum4, sum5, sum6, sum7, sum8]
            print((list1[0]))
            if(count==10):
                print("Match draw")
                exit()
            if(count%2==0):
                print("Player1 turn")
            else:
                print("Player2 turn")
            for i in list1:

                if (i == 15):
                    print("Element Exists and you win")
                    if(count%2==0 and play == 1):
                        print("Player2 wins")
                        # exit()
                        # goto check
                    elif(count%2==0 and play == 2):
                        print("Player1 wins")
                        # exit()
                        # goto check
                    elif (count % 2 != 0 and play == 2):
                        print("Player2 wins")
                        # exit()
                        # goto check
                    elif (count % 2 != 0 and play == 1):
                        print("Player1 wins")
                        # exit()
                        # goto check
    elif(game==2):
        print("Goodbye")
        exit()
    else:
        print("Enter valid choice")
```

References

[1] Balagusuamy, *C -Programming* McGraw-Hill, 4th ed.,2001.
https://www.pdfdrive.com/let-us-c-e33408389.html

[2] Tutorial point
https://www.tutorialspoint.com/cprogramming/

[3] W3School.com
https://www.w3schools.in/c-tutorial/

[4] Fresh2Refresh
https://fresh2refresh.com/c-programming/