Assignment 8

ELP - 718 Telecommunication Software Laboratory

Murali Krishnan K H

2019JTM2680

2019-2021

A report presented for the assignment on

Python and Github

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.
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
010101110100101
Sample Output
Parity bit data : 0101011101001011
Transmitting data: 01001011101000100110101

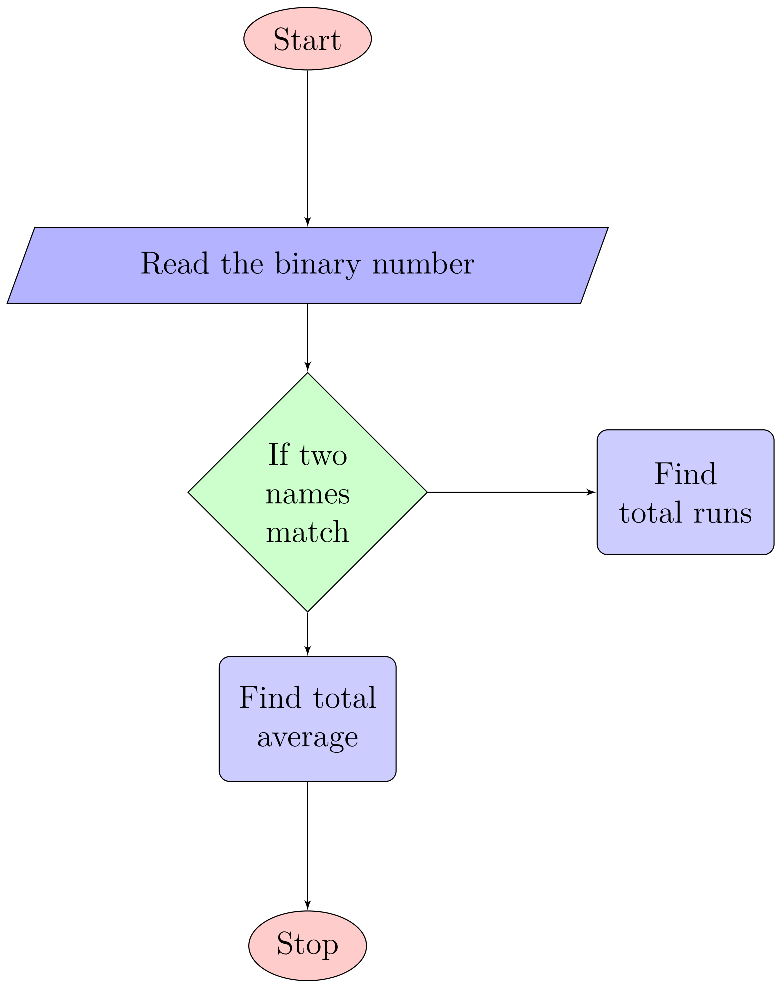
1.2 Assumptions

- We are verifying for odd parity. i.e., the number of ones should be odd
- The message after parity check should be encoded
- The message after encoding should be provided with an end flag

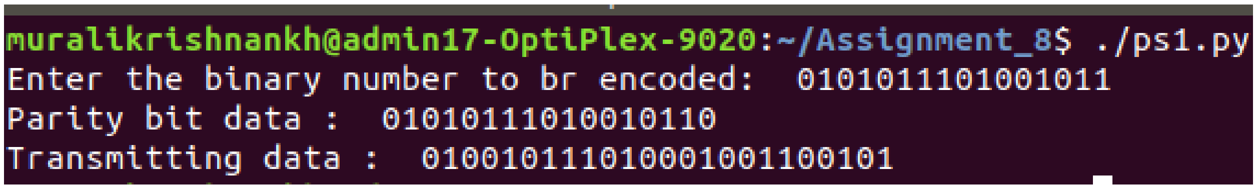
1.3 Algorithm and Implementation

- Read the input binary number
- check the number of one's for that number
- If number of one's are odd then add a 0 to the end of the number.
- If number of one is even add 1 to the end of the number.
- After adding the parity, check for pattern 010 for encoding the message.
- For 010 pattern replace it with 0100.
- At the end for the encoded message add 0101

1.4 Program Structure



1.5 Screenshots



```
muralikrishnankh@admin17-OptiPlex-9020:~/Assignment_8$ ./ps1.py
Enter the binary number to br encoded:  0101011101001011
Parity bit data :  01010111010010110
Transmitting data :  010010111010001001100101
```

Figure 1: Output for PS1 showing both parity and final message

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

Constraints: 1¡=Position¡=9 1¡=Number¡=9

Terminal: Print âĂŸWelcome to the Game!âĂŹ. Print whether it is Player 1âĂŹs or Player 2âĂŹs 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.

Sample Output:

Welcome to the Game! Player 1âĂŹs chance Enter the position and number to be entered: 5,3

3

Player 2âĂŹs chance Enter the position and number to be entered: 7,4

3

4

... Continue till game ends Note âĂŞ Must use at least one User Defined Function.

2.2 Assumptions

* Player A should give numbers 1,3,5,7,9 and B should give number 2,4,6,8
* The first player should enter only odd numbers
* One number should be entered only once
* After a position is given an input another player cannot replace it.

2.3 Algorithm and Implementation

- Initialize the board with positions of a list
- Take input from player A and B
- A should give odd inputs and B should give even inputs
- check horizontally, vertically and diagonally for sum to be 15
- If sum is equal to 15 the player who entered that number wins
- After filling the entire board if none of the combinations give 15 then the game is draw

2.4 Screenshots

3 Appendix

3.1 Code for ps1

```
#!/usr/bin/python3
#Reading the number
bin = input("Enter the binary number to br encoded:     ")
#Splitting the number into digits and storing it to a list
digits = [int(d) for d in str(bin)]
count=0
#counting the number of ones to implement parity check
for i in digits:
        if i==1:
                count+=1
#checking odd parity
if (count%2)==0:
        digits.append(int(1))
else:
        digits.append(int(0))
#converting the splitted list back to string
parity = ("".join(map(str, digits)))
#Output for fisrt part
print ("Parity bit data : ", parity)
#Encodind the message by adding 0 to the patter 010
temp = parity.replace("010","0100")
#Adding  string 0101 is used as the bit string or flag to indic
final = temp + "0101"
print ("Transmitting data : ", final)
```

3.2 Code for ps2

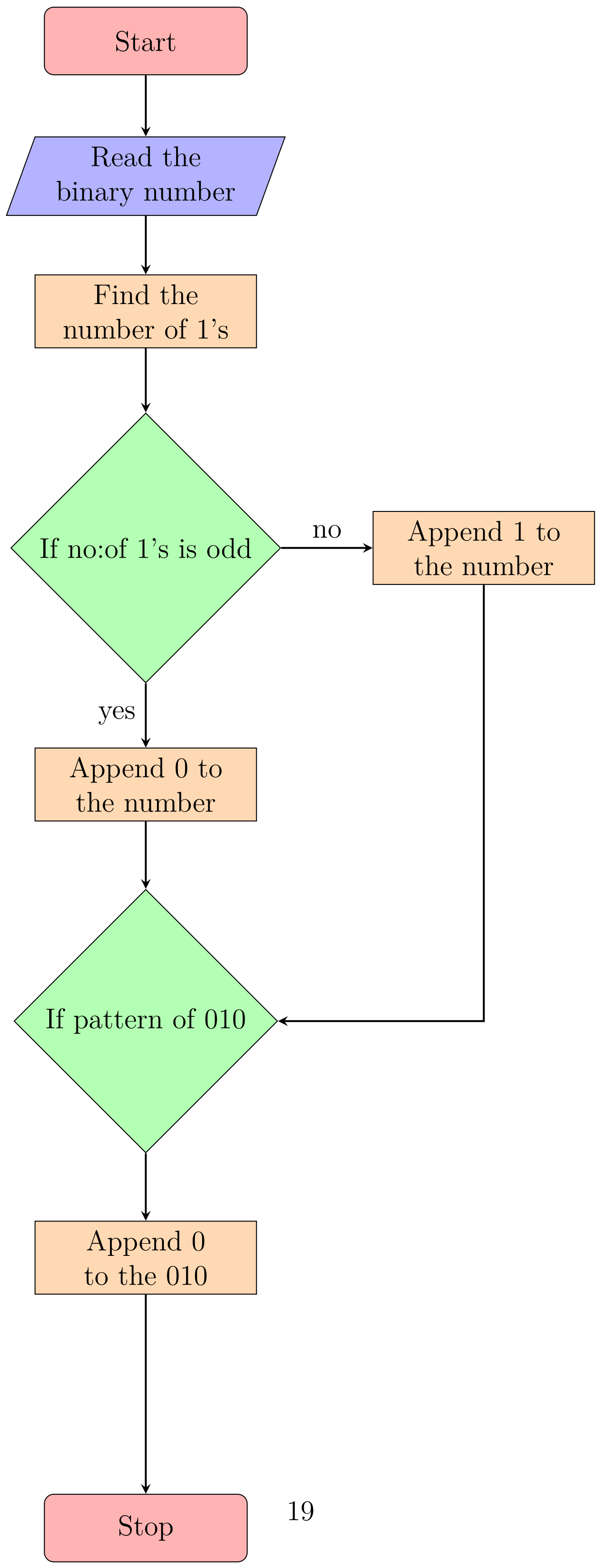
```
#!/usr/bin/python3
#The sys function is used for exiting program
import sys
#The below functios are declared to format the text(like adding
import shutil
columns = shutil.get_terminal_size().columns
def prRed(skk): print("\033[91m {}\033[00m" .format(skk))
def prGreen(skk): print("\033[92m {}\033[00m" .format(skk))
def prYellow(skk): print("\033[93m {}\033[00m" .format(skk))
def prCyan(skk): print("\033[96m {}\033[00m" .format(skk))
# Creating a function to display board
def board ():
        print ("\t\t\t\t\t\t\t\t\t\t\t\t'"--------------')
        print ("\t\t\t\t\t\t\t\t\t\t\t\t"'|' ,index[0],'|',inde
        print ("\t\t\t\t\t\t\t\t\t\t\t\t'"--------------')
        print ("\t\t\t\t\t\t\t\t\t\t\t\t"'|' ,index[3],'|',inde
        print ("\t\t\t\t\t\t\t\t\t\t\t\t'"--------------')
        print ("\t\t\t\t\t\t\t\t\t\t\t\t"'|' ,index[6],'|',inde
        print ("\t\t\t\t\t\t\t\t\t\t\t\t'"--------------')
#Function to check success
def check (index):
#Checking whether all three elements are inserted and whether t
        if index[0] != " " and index[1] != " " and index[2] !=
                if index[0] + index[1] + index[2] == 15:
                        return 1
        elif index[3] != " " and index[4] != " " and index[5] !=
                if index[3] + index[4] + index[5] == 15:
                        return 1
        elif index[6] != " " and index[7] != " " and index[8] !=
                if index[6] + index[7] + index[8] == 15:
                        return 1
        if index[0] != " " and index[4] != " " and index[8] !=
                if index[0] + index[4] + index[8] == 15:
                        return 1
        if index[0] != " " and index[3] != " " and index[6] !=
                if index[0] + index[3] + index[6] == 15:
                        return 1
        if index[6] != " " and index[4] != " " and index[2] !=
                if index[6] + index[4] + index[2] == 15:
                        return 1
        if index[2] != " " and index[5] != " " and index[8] !=
                if index[2] + index[5] + index[8] == 15:
                        return 1
        if index[1] != " " and index[4] != " " and index[7] !=
                if index[1] + index[4] + index[7] == 15:
                        return 1
        else:
                return 0
#main program starts here
prRed("TIC-TAC-TOE".center(columns))
prGreen("Welcome to the Game!".center(columns))
#Game instructions
prCyan("The index numbers and the index for game is indicated b
prCyan("Player A has numbers 1,3,5,7,9" .center(columns))
prCyan("Player B has numbers 2,4,6,8" .center(columns))
# index indicates the positions of board
index = [1,2,3,4,5,6,7,8,9]
board()
#Game starts
prCyan("Let's begin the Game!!!" .center(columns))
index = [" "," "," "," "," "," "," "," "," "]
board()
while True:
        #Accepting the input
        prCyan ("Player A, Enter your number and positon")
        num1 = int(input("Number : "))
        pos1 = int(input("Position : "))
#checking whether position is between 1-9
        if num1>0 and num1<10 and pos1>0 and pos1<10:
#checking whether odd number for player A
                if num1%2 != 0:
                        index[pos1-1] = num1
                        break
                else:
                        prRed ("Invalid input for Player A. Ple
        else:
                prRed ("The number/position entered is invalid.
#Displaying the board
board()
while True:
        prCyan ("Player B, Enter your number and positon")
        num2 = int(input("Number : "))
```

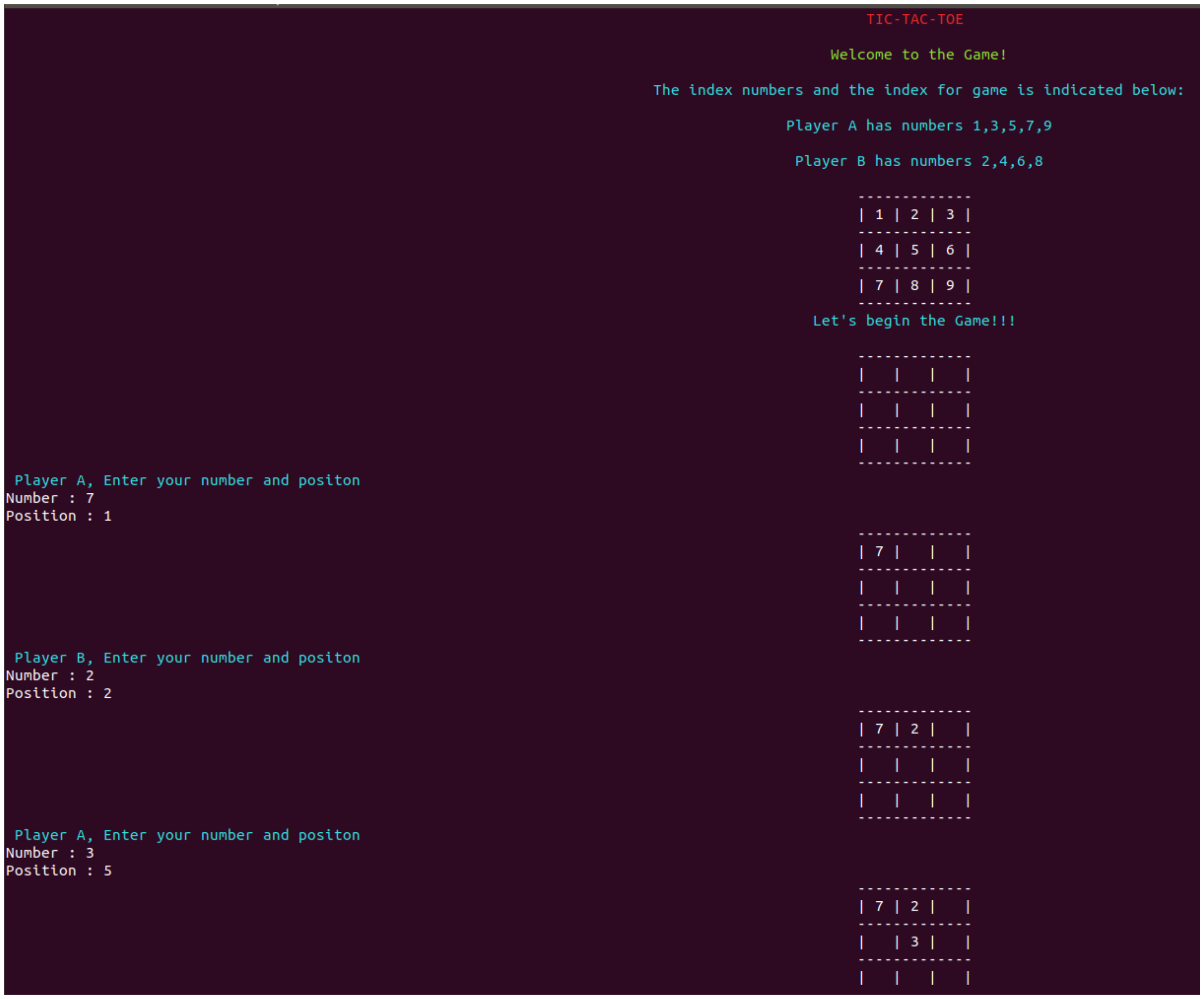
```
          pos2 = int(input("Position : "))
#checking whether odd number for player A
          if num2>0 and num2<10 and pos2>0 and pos2<10:
#checking whether it is an already entered postion or number
                    if num2 != num1 and pos2 != pos1:
#checking whether even number for player B
                              if num2%2 == 0:
                                        index[pos2-1] = num2
                                        break
                              else:
                                        prRed ("Invalid input for Playe
                    else:
                              prRed ("Invalid input. Already occupied
          else:
                    prRed ("The number/position entered is invalid.
board()
while True:
          prCyan ("Player A, Enter your number and positon")
          num3 = int(input("Number : "))
          pos3 = int(input("Position : "))
          if num3>0 and num3<10 and pos3>0 and pos3<10:
                    if num3 != num1 and num3 != num2 and pos3 != po
                              if num3%2 != 0:
                                        index[pos3-1] = num3
                                        break
                              else:
                                        prRed ("Invalid input for Playe
                    else:
                              prRed ("Invalid input. Already occupied
          else:
                    prRed ("The number/position entered is invalid.
board()
#checking for success
i = check(index)
if i == 1:
          prGreen("\nCONGRATULATIONS!!!! PLAYER A WINS!!!" .cente
          sys.exit()
while True:
          prCyan ("Player B, Enter your number and positon")
          num4 = int(input("Number : "))
          pos4 = int(input("Position : "))
          if num4>0 and num4<10 and pos4>0 and pos4<10:
                if num4 != num1 and num4 != num2 and num4 != nu
                        if num4%2 == 0:
                                index[pos4-1] = num4
                                break
                        else:
                                prRed ("Invalid input for Playe
                else:
                        prRed ("Invalid input. Already occupied
        else:
                prRed ("The number/position entered is invalid.
board()
i = check(index)
if i == 1:
        prGreen("\nCONGRATULATIONS!!!! PLAYER B WINS!!!" .cente
        sys.exit()
while True:
        prCyan ("Player A, Enter your number and positon")
        num5 = int(input("Number : "))
        pos5 = int(input("Position : "))
        if num5>0 and num5<10 and pos5>0 and pos5<10:
                if num5 != num1 and num5 != num2 and num5 != nu
                        if num5%2 != 0:
                                index[pos5-1] = num5
                                break
                        else:
                                prRed ("Invalid input for Playe
                else:
                        prRed ("Invalid input. Already occupied
        else:
                prRed ("The number/position entered is invalid.
board()
i = check(index)
if i == 1:
        prGreen("\nCONGRATULATIONS!!!! PLAYER A WINS!!!" .cente
        sys.exit()
while True:
        prCyan ("Player B, Enter your number and positon")
        num6 = int(input("Number : "))
        pos6 = int(input("Position : "))
        if num6>0 and num6<10 and pos6>0 and pos6<10:
                if num6 != num1 and num6 != num2 and num6 != nu
                        if num6%2 == 0:
```

```
                                        index[pos6-1] = num6
                                        break
                                else:
                                        prRed ("Invalid input for Playe
                        else:
                                prRed ("Invalid input. Already occupied
                else:
                        prRed ("The number/position entered is invalid.
        board()
        i = check(index)
        if i == 1:
                prGreen("\nCONGRATULATIONS!!!! PLAYER B WINS!!!" .cente
                sys.exit()
        while True:
                prCyan ("Player A, Enter your number and positon")
                num7 = int(input("Number : "))
                pos7 = int(input("Position : "))
                if num7>0 and num7<10 and pos7>0 and pos7<10:
                        if num7 != num1 and num7 != num2 and num7 != nu
                                if num7%2 != 0:
                                        index[pos7-1] = num7
                                        break
                                else:
                                        prRed ("Invalid input for Playe
                        else:
                                prRed ("Invalid input. Already occupied
                else:
                        prRed ("The number/position entered is invalid.
        board()
        i = check(index)
        if i == 1:
                prGreen("\nCONGRATULATIONS!!!! PLAYER A WINS!!!" .cente
                sys.exit()
        while True:
                prCyan ("Player B, Enter your number and positon")
                num8 = int(input("Number : "))
                pos8 = int(input("Position : "))
                if num6>0 and num6<10 and pos6>0 and pos6<10:
                        if num8 != num1 and num8 != num2 and num8 != nu
                                if num8%2 == 0:
                                        index[pos8-1] = num8
                                        break
```

```
                    else:
                            prRed ("Invalid input for Playe
            else:
                    prRed ("Invalid input. Already occupied
    else:
            prRed ("The number/position entered is invalid.
board()
i = check(index)
if i == 1:
        prGreen("\nCONGRATULATIONS!!!! PLAYER B WINS!!!" .cente
        sys.exit()
while True:
        prCyan ("Player A, Enter your number and positon")
        num9 = int(input("Number : "))
        pos9 = int(input("Position : "))
        if num9>0 and num9<10 and pos9>0 and pos9<10:
                if num9 != num1 and num9 != num2 and num9 != nu
                        if num9%2 != 0:
                                index[pos9-1] = num9
                                break
                        else:
                                prRed ("Invalid input for Playe
                else:
                        prRed ("Invalid input. Already occupied
        else:
                prRed ("The number/position entered is invalid.
board()
i = check(index)
if i == 1:
        prGreen("\nCONGRATULATIONS!!!! PLAYER A WINS!!!" .cente
        sys.exit()
else:
        prGreen("\nGAME DRAWN!!!" .center(columns))
```

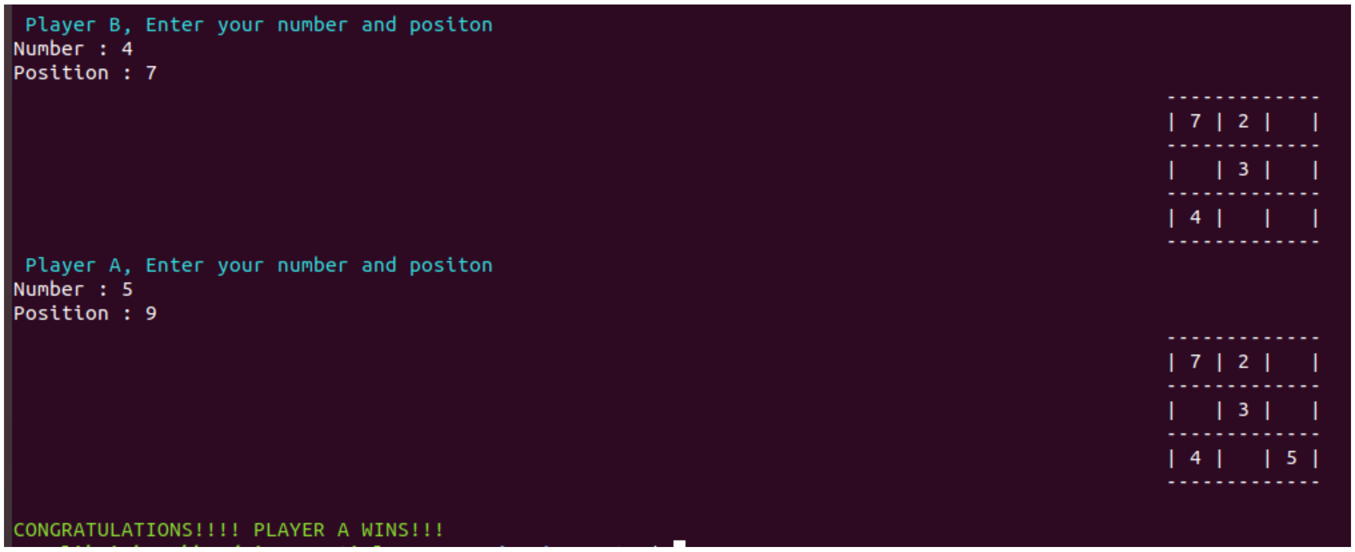
References


Start
Read the binary number
Find the number of 1's
If no:of 1's is odd
no
Append 1 to the number
yes
Append 0 to the number
If pattern of 010
Append 0 to the 010
Stop



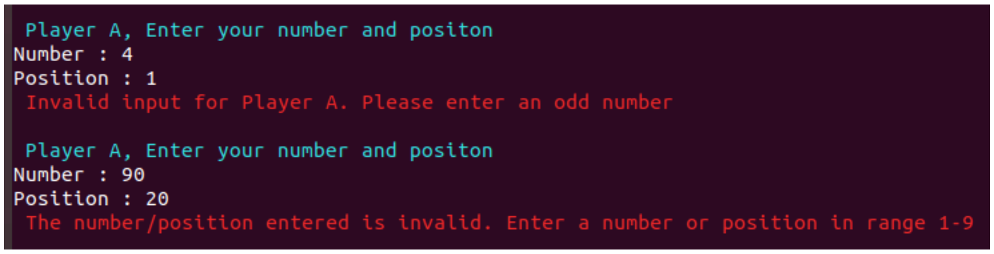
```
                                                            TIC-TAC-TOE

                                                        Welcome to the Game!

                                        The index numbers and the index for game is indicated below:

                                                    Player A has numbers 1,3,5,7,9

                                                     Player B has numbers 2,4,6,8

                                                            -------------
                                                            | 1 | 2 | 3 |
                                                            -------------
                                                            | 4 | 5 | 6 |
                                                            -------------
                                                            | 7 | 8 | 9 |
                                                            -------------
                                                        Let's begin the Game!!!

                                                            -------------
                                                            |   |   |   |
                                                            -------------
                                                            |   |   |   |
                                                            -------------
                                                            |   |   |   |
                                                            -------------
 Player A, Enter your number and positon
Number : 7
Position : 1

                                                            -------------
                                                            | 7 |   |   |
                                                            -------------
                                                            |   |   |   |
                                                            -------------
                                                            |   |   |   |
                                                            -------------
 Player B, Enter your number and positon
Number : 2
Position : 2

                                                            -------------
                                                            | 7 | 2 |   |
                                                            -------------
                                                            |   |   |   |
                                                            -------------
                                                            |   |   |   |
                                                            -------------
 Player A, Enter your number and positon
Number : 3
Position : 5

                                                            -------------
                                                            | 7 | 2 |   |
                                                            -------------
                                                            |   | 3 |   |
                                                            -------------
                                                            |   |   |   |
```

Figure 2: Output file of PS2 showing game



```
 Player B, Enter your number and positon
Number : 4
Position : 7
                                                                                              -------------
                                                                                              | 7 | 2 |   |
                                                                                              -------------
                                                                                              |   | 3 |   |
                                                                                              -------------
                                                                                              | 4 |   |   |
                                                                                              -------------

 Player A, Enter your number and positon
Number : 5
Position : 9
                                                                                              -------------
                                                                                              | 7 | 2 |   |
                                                                                              -------------
                                                                                              |   | 3 |   |
                                                                                              -------------
                                                                                              | 4 |   | 5 |
                                                                                              -------------

CONGRATULATIONS!!!! PLAYER A WINS!!!
```

Figure 3: Output file of PS2 showing game result



```
 Player A, Enter your number and positon
Number : 4
Position : 1
 Invalid input for Player A. Please enter an odd number

 Player A, Enter your number and positon
Number : 90
Position : 20
 The number/position entered is invalid. Enter a number or position in range 1-9
```

Figure 4: Output file of PS2 showing exception handling