INDIAN INSTITUTE OF TECHNOLOGY,DELHI

TELECOM SOFTWARE LAB

ASSIGNMENT NUMBER-6

NAME: VAIBHAV NIGAM
ENTRY NO.:2017JTM2023

DATE: 12 SEPTEMBER, 2017

CONTENT

1. Problem Statements
2. Assumptions
3. Algorithm and Implementation
4. Input and output format
5. Screenshots
6. References

Problem Statements

0.1 Problem Statement 1

Create a pytho program to check the parity of the entered binary string.The parity bit should have a value 1 if the number of 1's in the bit string is even and have a value 0 otherwise, i.e odd parity check.The string 0101 must be used as the bit string or flag to indicate the end of the frame. The bit stuffing rule is to insert a 0 after each appearance of 010 in the original data. In addition, if the frame ends in 01, a 0 would be stuffed after the 1st 0 in the actual terminating string 0101.

0.2 Problem Statement 2

Create a 3x3 Numeric tic-tac-toe but use numbers 1 to 9 instead of Xs and Os.One player should play with the odd numbers (1, 3, 5, 7, 9) and other player with the even numbers (2,4,6,8).The player who scores a sum of 15 in any of the line i.e vertical , horizontal or diagnol first will be the winnner. The game should start with the player with odd numbers.

Assumptions

0.3 PS1

1.The user must enter the string with binary bits only.
2. The string can be of any length.

0.4 PS2

1.A player can player with either even or odd numbers.
2.Player playing with odd numbers must start the game.
3. All numbers can be used only once.

Algorithm and implementation

0.5 Problem Statement1

1.The user enters the bit string.
2.Add the digits of the string.
3.Divide the sum with 2.
4.If the remainder is 0 i.e even, then add a 1 in the end of the string. 5. else it is odd, then add a 0 at the end.
6.Then add the flag i.e 0101 at the end of the string.

0.6 Problem Statement2

1.A list having nine positons is created.
2.choose the player1 or player2.
3.The position where the player wants to enter the digit is entered.
4.the entered position choosed by th player is replace by the the number entered.
5.If the sum in any line becomes equal to 15, the player wins.

Input and Output Format

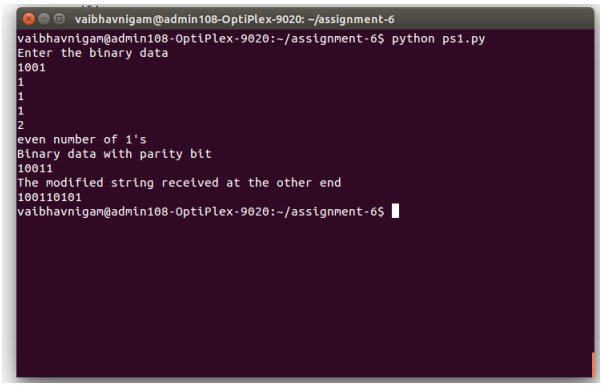
0.7 Problem Statement1

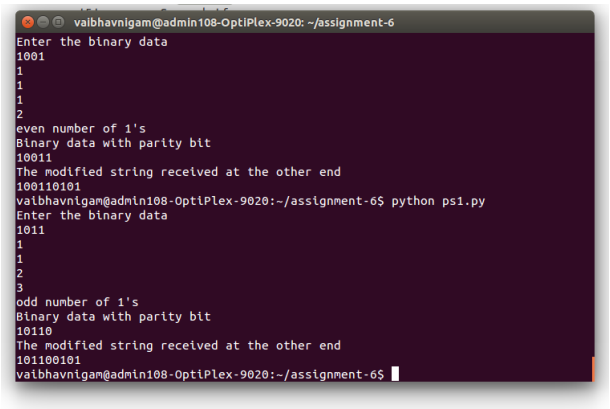
1.Binary bit string is entered by the user at the run time.
2.Output is also a bit string.

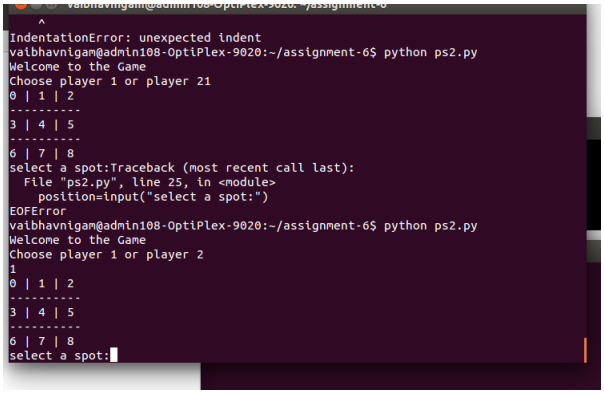
0.8 Problem Statement2

1.Position is between 1 to 9.
2.Numbers to be entered are also between 1 to 9.

Screenshots







References

[1]www.docs.python.org

[2]Google python classes

Annexure

```
#!/home/assignment-6/ps1.py
```

```
###### this is the first .py file ###########

####### write your code here ##########

binary_data = input("Enter the binary data\n")
flag ='0101'
even= '1'
even_parity= int(str(binary_data)+ str(even))
even_parity_flag= int(str(binary_data)+ str(even)+ str(flag))
odd= '0'
odd_parity= int(str(binary_data)+ str(odd))
odd_parity_flag= int(str(binary_data)+ str(odd)+ str(flag))

def digitsum(binary_data):
    total=0
    for letter in str(binary_data):
        total+=int(letter)
        print(total)

    return total;



total=digitsum(binary_data)

if total%2 == 0:
    print("even number of 1's")
    print("Binary data with parity bit")
    print(even_parity)
    print('The modified string received at the other end')
    print(even_parity_flag)
else:
    print("odd number of 1's")
    print("Binary data with parity bit")
    print(odd_parity)
    print('The modified string received at the other end')
    print(odd_parity_flag)
```

```
#!/home/assignment-6/ps2.py
```

```
###### this is the second .py file ###########

####### write your code here ##########
#the game board

print("Welcome to the Game")

player=input("Choose player 1 or player 2\n")

gameboard= [0,1,2,
            3,4,5,
            6,7,8]

def showboard():
    print gameboard[0],'|',gameboard[1],'|',gameboard[2]
    print "----------"
    print gameboard[3],'|',gameboard[4],'|',gameboard[5]
    print "----------"
    print gameboard[6],'|',gameboard[7],'|',gameboard[8]


showboard()

while True:
    position=input("select a spot:")
    position = int (position)
```