Assignment- 8

ELP - 718 Telecom Software Laboratory

Ch Krishna Chaitanya
2019JTM2674
2019-21

A report on Python Basics

Bharti School of
Telecommunication Technology and Management
IIT Delhi
India

September 18, 2019

Contents

1 Problem Statement -1 **2**

1.1 Problem Satement 2

1.2 Algorithm and Implementation 2

1.3 Flowchart 3

1.4 Input and Output format 3

1.4.1 Input Format 3

1.4.2 Output Format 3

1.5 Test Cases 4

1.5.1 Input1 4

1.5.2 Output1 4

1.6 Screenshots 4

2 Problem Statement -2 **4**

2.1 Problem Satement 4

2.2 Algorithm and Implementation 4

2.3 Flowchart 5

2.4 Test Cases 6

2.4.1 Constraints 6

2.4.2 Input 6

2.4.3 Output 6

2.5 Screenshots 7

Appendices **7**

References **13**

Objective Statement

To Test our understanding of python basics including lists and functions.

1 Problem Statement -1

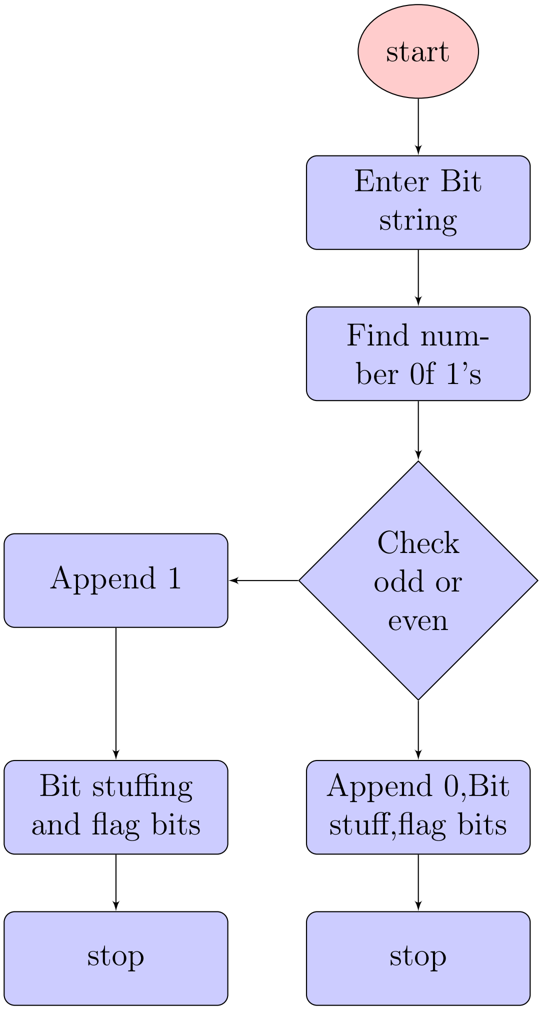
To write a python program to find simplest way of error detection and to append a single bit, called a parity check, to a string of data bits

1.1 Problem Satement

1.2 Algorithm and Implementation

- Enter Input
- Check Parity
- Print data after parity check
- Find substring
- Append 0 after substring
- Append 0101 to indicate end of frame

1.3 Flowchart



1.4 Input and Output format

1.4.1 Input Format

Enter binary bit data that has to be transmitted.

1.4.2 Output Format

Print binary bit data with parity bit.
Print the modified string that is to be transmitted

1.5 Test Cases

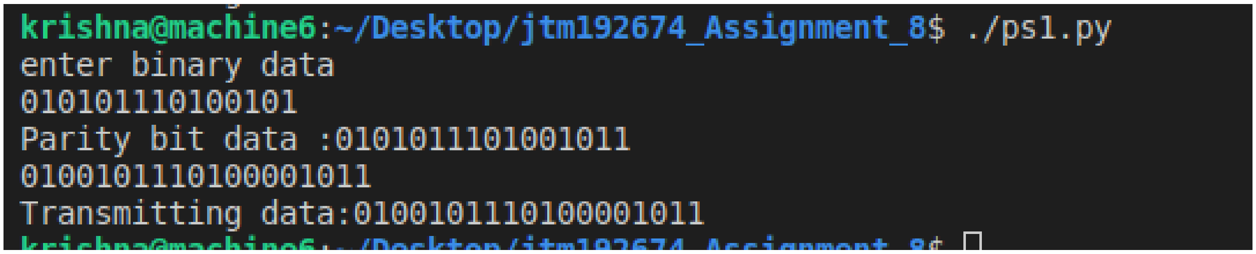
1.5.1 Input1

010101110100101

1.5.2 Output1

Parity bit data : 0101011101001011
Transmitting data: 01001011101000100110101

1.6 Screenshots



```
krishna@machine6:~/Desktop/jtm192674_Assignment_8$ ./ps1.py
enter binary data
010101110100101
Parity bit data :0101011101001011
01001011101000001011
Transmitting data:01001011101000001011
krishna@machine6:~/Desktop/jtm192674_Assignment_8$ 
```

2 Problem Statement -2

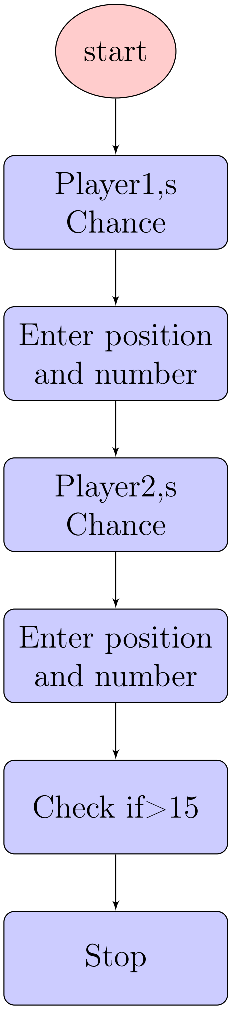
2.1 Problem Satement

To write a python program for a 3X3 tic tac toe.

2.2 Algorithm and Implementation

- Check if player want to enter odd number
- Player1 enters choice
- Enter number and position between 0-9
- Player2 enters choice
- Enter number and position between 0-9
- Check if sum in a row or column or diagonal > 15
- If > 15, Gameover

2.3 Flowchart



2.4 Test Cases

2.4.1 Constraints

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

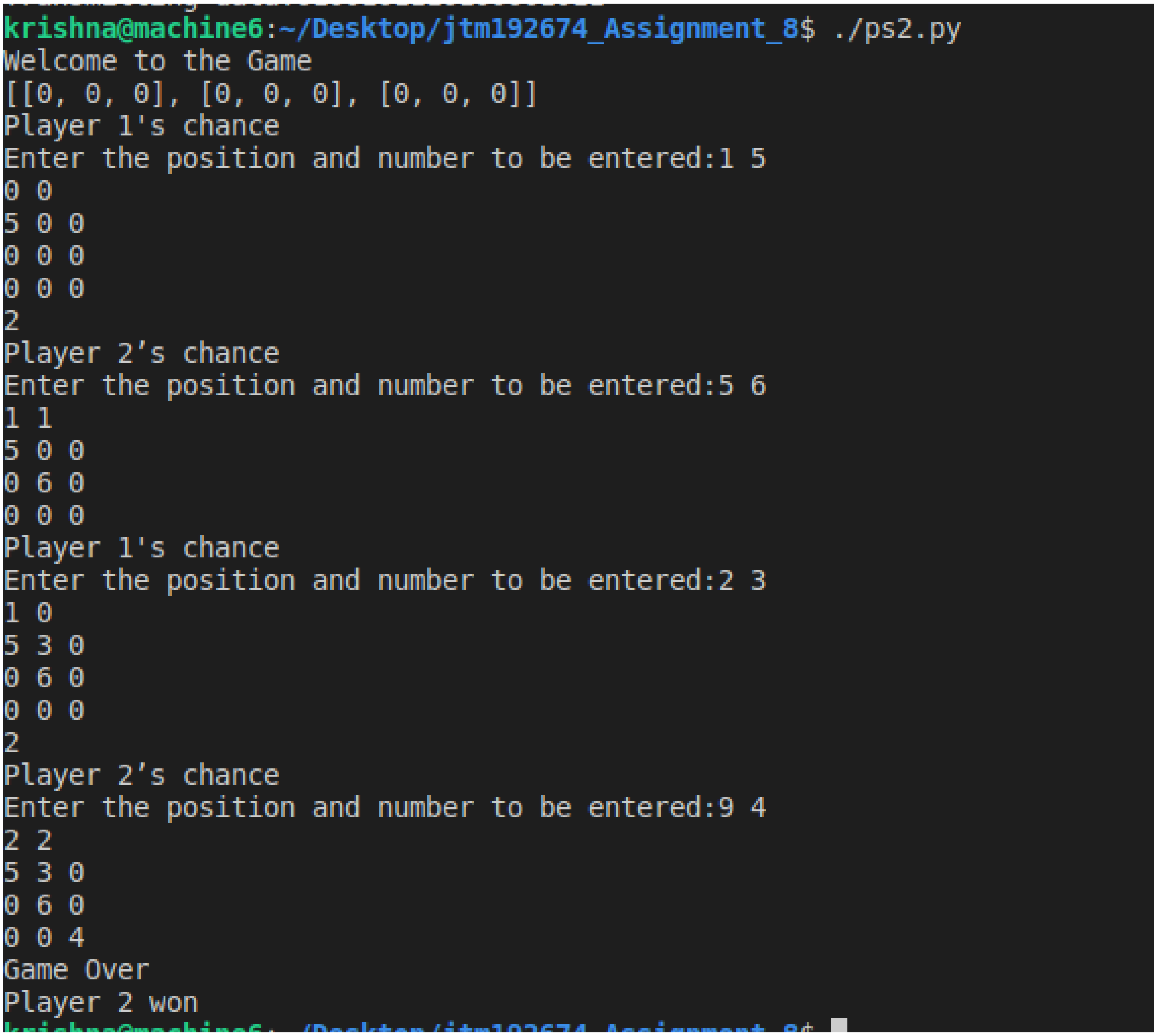
2.4.2 Input

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.

2.4.3 Output

```
Welcome to the Game
Player 1’s chance
Enter the position and number to be entered:1 5 0 0
5 0 0
0 0 0
0 0 0
2
Player 2’s chance
Enter the position and number to be entered:5 6
1 1
5 0 0
0 6 0
0 0 0
Continue
```

2.5 Screenshots



```
krishna@machine6:~/Desktop/jtm192674_Assignment_8$ ./ps2.py
Welcome to the Game
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
Player 1's chance
Enter the position and number to be entered:1 5
0 0
5 0 0
0 0 0
0 0 0
2
Player 2’s chance
Enter the position and number to be entered:5 6
1 1
5 0 0
0 6 0
0 0 0
Player 1's chance
Enter the position and number to be entered:2 3
1 0
5 3 0
0 6 0
0 0 0
2
Player 2’s chance
Enter the position and number to be entered:9 4
2 2
5 3 0
0 6 0
0 0 4
Game Over
Player 2 won
krishna@machine6:~/Desktop/jtm192674_Assignment_8$
```

Appendices

Problem 1

code:

```
#!/usr/bin/python3
import re
# taking binary data
```

```
print("enter binary data")
binary_string=input()
#print(string)
count = 0
#finding length of string
l=len(binary_string)
for i in range(0,l):
    if binary_string[i]==’1’:     #counting number of 1’s
        count+=1
#parity checking
if count%2==1:
    print("odd")
    binary_string += ’0’
else:
    binary_string += ’1’

#printing binary data after parity check
print("Parity bit data :"+binary_string)
sub_string=’010’

#finding indices after checking ’010’
match_index = [k.start() for k in re.finditer(sub_string,binary_string)]

#convert string to list
list_string=list(binary_string)
for x in match_index:
    list_string[x+len(sub_string):x+len(sub_string)]=’0’


#print(match_index)
#print(list_string)

#convet list to string
transmit_string=’’.join(map(str,list_string))
print(transmit_string)
transmit_string+=’0101’
#printing binary data after parity check
print("Transmitting data:"+transmit_string)
```

Problem 2

code:

```
#!/usr/bin/python3
import math
#tic tac code

#function for calculation
def evaluate(matrix):
    sum_row1=0
    sum_col2=0
    sum_col1=0
    sum_col3=0
    sum_row2=0
    sum_dia1=0
    sum_dia2=0
    sum_row3=0
    sum_row1+=matrix[0][0]+matrix[0][1]+matrix[0][2]  #finding sum of rows
    sum_row2+=matrix[1][0]+matrix[1][1]+matrix[1][2]
    sum_row3+=matrix[2][0]+matrix[2][1]+matrix[2][2]

    sum_col1+=matrix[0][0]+matrix[1][0]+matrix[2][0]  #finding sum of cols
    sum_col2+=matrix[0][1]+matrix[1][1]+matrix[2][1]
    sum_col3+=matrix[0][2]+matrix[1][2]+matrix[2][2]

    sum_dia1+=matrix[0][0]+matrix[1][1]+matrix[2][2] #finding sum of diagonals
    sum_dia2+=matrix[0][2]+matrix[1][1]+matrix[2][0]

    #checking condition
    if sum_col1>15 or sum_col2>15 or sum_col3>15 or sum_row1>15 or sum_row2>15 or sum_ro
        print("Out of Range")
        print("Re enter")
        return 3
    elif sum_col1==15 or sum_col2==15 or sum_col3==15 or sum_row1==15 or sum_row2==15 or
        print("Game Over")
        return 1
    else:
        return 2


#function for taking input
def ticTakToe():
    matrix=[]
    change=3
    for col in range(1,4):
```

```
    row=[]
    for r in range(1,4):
        row.append(0)
    matrix.append(row)
print(matrix)

while 1 :
    print("Player 1’s chance")
    pos,num=[int(x) for x in input("Enter the position and number to be entered:").s
    #exception handling
    if(num>9):
        print("Error: Range of number is 0-9")
        print("Re enter")
        continue
    #exception handling
    if(pos>9):
        print("Error: Range of position is 0-9")
        print("Re enter")
        continue
    #exception handling
    if(num%2==0):
        print("Invalid input. Enter a odd number")
        continue

    #forming a matrix
    if pos<4:
        y=math.floor((pos-1)/3)
        x=(pos-1)%3
        print(x,y)
        matrix[y][x]=num
    elif pos>3 and pos<7:
        y=math.floor((pos-1)/3)
        x=(pos-1)%3
        print(x,y)
        matrix[y][x]=num
    elif pos>6 and pos<10:
        y=math.floor((pos-1)/3)
        x=(pos-1)%3
        print(x,y)
        matrix[y][x]=num

    for mat in matrix:
```

```
    print(*mat)

flag=evaluate(matrix)
print(flag)
if flag==1:
    print("Player1 won")
    break
if flag==3:
    matrix[y][x]=0
    continue

while change==3:

    print("Player 2's chance")
    pos,num=[int(x) for x in input("Enter the position and number to be entered:
     #exception handling
    if(num>9):
        print("Error: Range of number is 0-9")
        print("Re enter")
        continue
     #exception handling
    if(pos>9):
        print("Error: Range of position is 0-9")
        print("Re enter")
        continue
     #exception handling
    if(num%2!=0):
        print("Invalid input. Enter a even number")
        continue
    #forming a matrix
    if pos<4:
        y=math.floor((pos-1)/3)
        x=(pos-1)%3
        print(x,y)
        matrix[y][x]=num
    elif pos>3 and pos<7:
        y=math.floor((pos-1)/3)
        x=(pos-1)%3
        print(x,y)
        matrix[y][x]=num
    elif pos>6 and pos<10:
        y=math.floor((pos-1)/3)
```

```
                x=(pos-1)%3
                print(x,y)
                matrix[y][x]=num

            for mat in matrix:
                print(*mat)

            flag2=evaluate(matrix)
            if flag2==2 or flag2==1:
                break
            print(flag)
            if flag2==3:
                matrix[y][x]=0
                change=3
                continue
        if flag2==1:
            print("Player 2 won")
            break



#main function
print("Welcome to the Game")
score=ticTakToe()
```

References

[1] Flowchart using Latex
Kjell Magne Fauske
http://www.texample.net/tikz/examples/simple-flow-chart/

[2] Python Basics
https://docs.python.org/3/

[3] Python Examples
https://www.geeksforgeeks.org/python-list/

[4] Git Hub
https://help.github.com/en/articles/fork-a-repo

[5] Python regular expressions
https://docs.python.org/3/howto/regex.html