TELECOMM. SOFTWARE LAB
ELP718
ASSIGNMENT NO- 6

ANKITA GUPTA
ENTRY NO.- 2017JTM2021
SEMESTER-I

September 12, 2017

BHARTI SCHOOL OF
TELECOMMUNICATION TECHNOLOGY AND
MANAGEMENT
IIT DELHI
INDIA

Contents

Problem Statement 1 2
.1 Algorithm and Implementation 2
.2 Input and output format 2
.3 Test Cases 3
.4 Screenshots 3
Problem Statement 2 4
.1 Assumptions 4
.2 Algorithm and Implementation 5
.3 Input and output format 5
.4 Test Cases 5
.5 Screenshots 5
References 6
Annexure 7

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 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 end of frame using bit oriented approach. It uses a special string of bits, called a flag for both idle fill and to indicate the beginning and the ending of frames. The string 0101 is 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.

.1 Algorithm and Implementation

- **Input the binary data from the user.**
- **check for the number of one's present in the data.**
- **If number of one's in data are even then append 1 at the end.**
- **Otherwise append zero at the end.**
- **Check this parity added data for bit stuffing.**
  - **If there is presence of 010 in the parity added data then stuff a '0' after 010.**
  - **Otherwise let the data as it is.**
- **If the parity added data ends with 01 then stuff '0' in that data otherwiswe let the data as it is.**
- **add the flag at the end of the bit stuffed data to transmit.**
- **print the modified data on the terminal.**

.2 Input and output format

Input format

Enter binary bit data which has to be transmitted.

Outpu format

Print binary bit data with parity bit. Print the modified string received at the other end.

.3 Test Cases

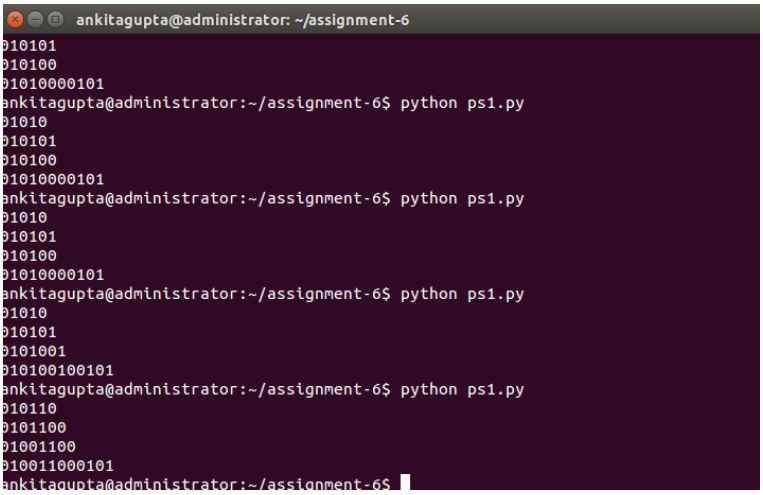
sample input
01010

sample output
010101
010100100101

sample input
010110

sample output
01001100
010011000101

.4 Screenshots



from 2017-09-12 16:14:37.png

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 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 start 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 cell might be necessary to complete a different line. Note – Line can be horizontal, vertical or diagonal

.1 Assumptions

- **1¡=Position¡=9**
- **1¡=Number¡=9**
- **1¡=Sum¡=15**

.2 Algorithm and Implementation

- **To design a tic tac toe game which is played between two players.**
- **The first player can choose only odd numbers from 1-9.**
- **Second player can only choose even numbers from 1-9.**
- **The player who will attain 15 or more points will be a winner and that game should be terminated there.**
- **count should be check vertical, horizontal and diagonal.**
- **Again ask if they want to continue game or not.**

.3 Input and output format

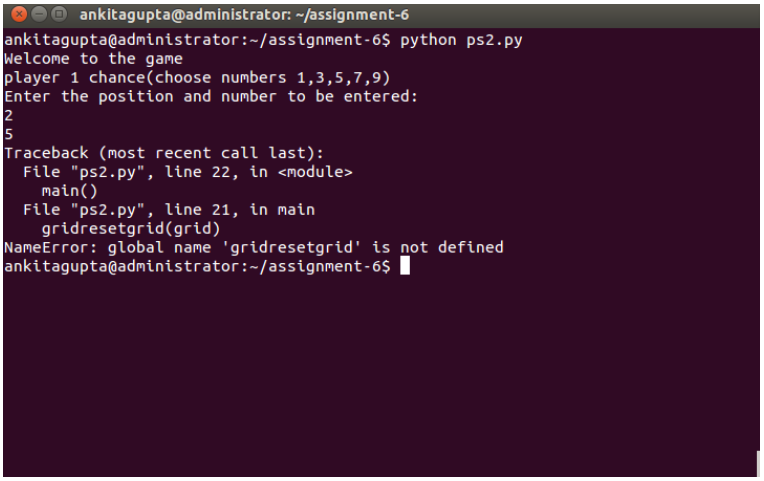
input Format

- **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 user.**
- **Show tic tac toe with data.**
- **Continue till the game gets draw or some player wins and show result.**
- **Ask user whether to continue for next game or exit.**

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

.4 Test Cases

.5 Screenshots

from 2017-09-12 16:46:59.png 

References

- **https://www.tutorialspoint.com/python**
- **http://www.sanfoundry.com/python-problems-solutions/**

Annexure

problem statement 1

```
###### this is the ps1 .py file ##########

n=raw_input()
count=0
count1=0
x=len(n)
#to check even parity
for i in range(0,x):
    if n[i]=='1':
        count=count+1

#to concatinate 1 if even parity

if count%2==0:
    n_parity= n[:x]+'1'

#concatinate 0 if odd parity

else:
    n_parity= n[:x]+'0'
print n_parity

x1=len(n_parity)

#bit stuffing after 010

str2='010'
for i in range(0,x):
    if n_parity[i:i+3]==str2:
        index=i+3

    n1_parity=n_parity[:index]+'0'+n_parity[index:]

print n1_parity

#to add 0 if the frame ends with 01

str1="01"
if n_parity[x1-2:]==str1:
    n1_parity=n1_parity[:len(n1_parity)]+'0'

#print n1_parity

#to add the flag at the end of frame

modified_string=n1_parity[:len(n1_parity)]+'0101'

print modified_string
```

problem statement 2

```
###### this is the ps2 .py file ##########

#function to print tic tac toe board

def print_board():
    for i in range(0,3):
        for j in range(0,3):
            print map[2-i][j]
            if j!=2:
                print "|"
        print ""

def playgame():

print "Welcome to the game"

print "player 1 chance(choose numbers 1,3,5,7,9)"
print "Enter the position and number to be entered:"
pos=int(raw_input())
num=int(raw_input())
grid=[]
gridresetgrid(grid)
```