Assignment No. 6

ELP - 718 Telecom Software Lab

Basic Python Programming

Submitted By:
Induru Sunil Reddy
2017JTM2768
Semester 1

Supervisor:
Prof. Subrat Kar

Bharti School
Indian Institute of Technology Delhi
September 12, 2017

Contents

1 Problem Statement

1.1 ps1

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.

1.2 ps2

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

2 Assumptions

2.1 ps1

Port no. on client side is 9000.

2.2 ps2

1 <=Position<= 9
1 <=Number<= 9
1 <=Sum<= 15

3 Input and OutPut Format

3.1 ps1

Input Format
Enter binary bit data which has to be transmitted.

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

3.2 ps2

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 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.

Sample Output:
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.

4 Program Structure

5 Algorithm and Implementation

5.1 ps1

Check Appendix.

6 Test Cases

6.1 ps1

Sample Input
01010

Sample Output
010101 0100100100101

7 Difficulties Faced

In Ps1:

1. Bit stuffing 2. ¿15 sum in tic tac toe problem

8 Screenshots

9 References

1. https://stackoverflow.com/questions/27491005/how-can-i-get-a-string-from-input-without-including-a-newline-using-fgets 2. https://stackoverflow.com/questions/23145997/how-do-you-keep-a-socket-connection-open-indefinitely-in-c

10 Appendix

10.1 ps1

ps1

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

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

## Parity Checking

sample = raw_input("Sample Bits :")
check = 0;

for i in sample
  if(i!="0" || i!="1"):
    break;
  else:
    if i="1":
      check=check+1
    else:
      continue
```

```
if (check/2 == 0)
  print("sample"+"1")
else
  print("sample"+"0")

## Bit Stuffing
temp = None
count = 0
output = None
for i in sample:
  temp = temp+i
  count = count+1
  if (count==3 and temp = "010"):
    output = output+temp+'0'
    temp=None
    count = 0
    i=i-2   #going back two spaces in the sample
    continue
  else:
    continue

#adding 0101 at the end of sample

length = len(sample)
lasttwo = None
for j in sample range(length-2,length):
  lasttwo=lasttwo+j
  if(lasttwo=="10"):
    output = output+'00101' #frame ends in 01, a 0 would be stuffed after the 1
    st 0 in the actual terminating string 0101.
  else:
    output = output+"0101"

print('output')
```

ps2
Having Ton of errors for this. Don't know why.