Assignment-6

ELP - 718 Telecom Software Laboratory

Ankit Dixit

2017JTM2769

semester-1

A report presented for the assignment

Developing logical skills to solve the given problem with the help of basics of Python

Bharti School Of

Telecommunication Technology and Management

IIT DELHI
India

September 12, 2017

Contents

1 Problem Statement-1

1.1 Problem statement

Write a code in python for displaying the message frame along with the parity checking i.e.odd parity check and then after the bit stuffing based on the rule given in the assumptions.

1.2 Assumptions

- **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 1s 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.

1.3 Program structure

- first part of the programme we take input from the user and store them
- then store them into a list and apply the logic to calculate the bits that is to be stuffed and parity bit.
- after the the calculation merge the list element that is append the same list with extra element.
- then finaaly show the output.

1.4 Algorithm and Implementation

- take no of bits in mesage from the user.
- then store them into a list.
- calculate the parity bit.
- stuff this into the same list.

- now check for the condition that is given and stuff the bit based on it.
- now finally print the list.

1.5 Input and Output format

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.

1.6 Test cases

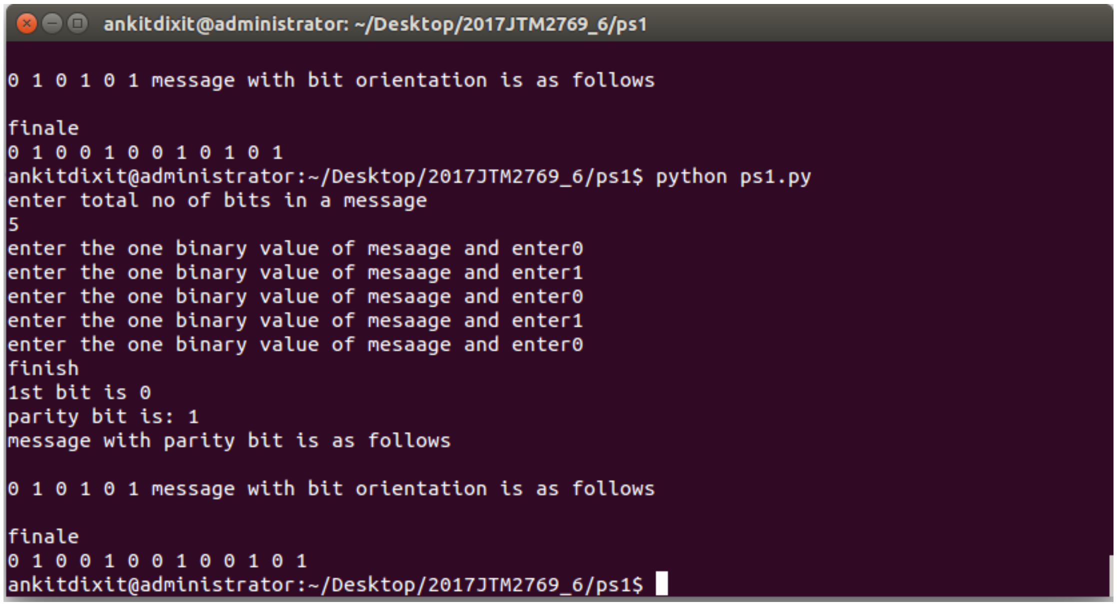
Sample Input
01010

Sample Output
010101
0100100100101

1.7 Difficulties /Issues faced

- over indexing.
- logic development.
- format error.
- syntsx error.

1.8 Screenshots



```
0 1 0 1 0 1 message with bit orientation is as follows

finale
0 1 0 0 1 0 0 1 0 1 0 1
ankitdixit@administrator:~/Desktop/2017JTM2769_6/ps1$ python ps1.py
enter total no of bits in a message
5
enter the one binary value of mesaage and enter0
enter the one binary value of mesaage and enter1
enter the one binary value of mesaage and enter0
enter the one binary value of mesaage and enter1
enter the one binary value of mesaage and enter0
finish
1st bit is 0
parity bit is: 1
message with parity bit is as follows

0 1 0 1 0 1 message with bit orientation is as follows

finale
0 1 0 0 1 0 0 1 0 0 1 0 1
ankitdixit@administrator:~/Desktop/2017JTM2769_6/ps1$ 
```

2 Problem Statement-2

2.1 Problem statement

3X3 Numeric Tic-Tac-Toe (Use numbers 1 to 9 instead of Xs and Os) 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

2.2 Assumptions

Constraints:

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

2.3 Program structure

- We define the users
- then after defineing the users take their preference and allot them the values
- check for the constraints at the time of input and then perform it.

2.4 Algorithm and Implementation

2.5 Input and Output format

- **Input format**
  input is in the form of numerics.
- **Output format**
  output is in the form of text that whether you wins or not.

2.6 Test cases

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

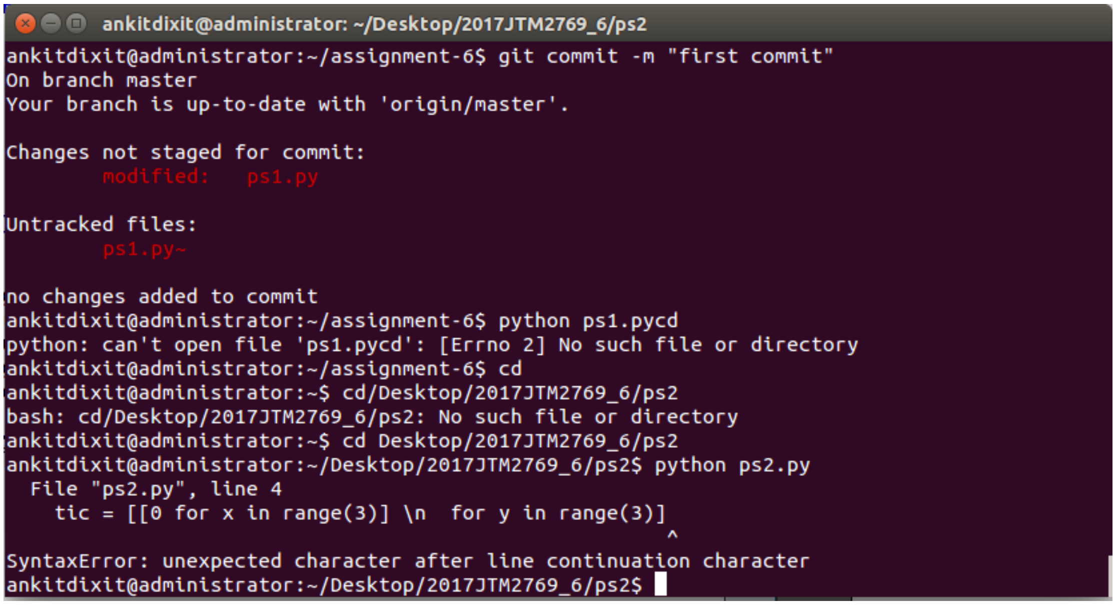
3
Player 2s chance
Enter the position and number to be entered: 7,4

3
4
... Continue till game ends
Note Must use at least one User Defined Function.

2.7 Difficulties /Issues faced

- over indexing.
- logic development.
- format error.
- syntsx error.

2.8 Screenshots



```
ankitdixit@administrator:~/assignment-6$ git commit -m "first commit"
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
        modified:   ps1.py

Untracked files:
        ps1.py~

no changes added to commit
ankitdixit@administrator:~/assignment-6$ python ps1.pycd
python: can't open file 'ps1.pycd': [Errno 2] No such file or directory
ankitdixit@administrator:~/assignment-6$ cd
ankitdixit@administrator:~$ cd/Desktop/2017JTM2769_6/ps2
bash: cd/Desktop/2017JTM2769_6/ps2: No such file or directory
ankitdixit@administrator:~$ cd Desktop/2017JTM2769_6/ps2
ankitdixit@administrator:~/Desktop/2017JTM2769_6/ps2$ python ps2.py
  File "ps2.py", line 4
    tic = [[0 for x in range(3)] \n  for y in range(3)]
                                                       ^
SyntaxError: unexpected character after line continuation character
ankitdixit@administrator:~/Desktop/2017JTM2769_6/ps2$
```

Appendix

3 References

- let us C.
- https://help.gnome.org/users/gedit/stable/
- http://fresh2refresh.com/c-programming/c-data-types/
- https://www.tutorialspoint.com/python/python$_lists.htmhttps : //help.github.com/articles/fork-a-repo/$
- https://stackoverflow.com/questions/11119632/bitwise-xor-of-hex-numbers-in-python
- http://www.physics.nyu.edu/pine/pymanual/html/chap3/chap3$_arrays.html$