Assignment-8

ELP-718 Telecom Software Laboratory

Satyam singh

2019JTM2691

2019-2021

A report for the assignment on Python Basics and Github

Bharti School Of

Telecommunication Technology And Management

IIT Delhi

India

Contents

1 Problem Statement-1

1.1 Problem Statement

- **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 the 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 the end of the frame using a bit-oriented approach. It uses a special string of bits, called a flag for both idle fills and to indicate the beginning and the ending of frames. The bit stuffing rule is to insert a 0 after each appearance of 010 in the original data. The string 0101 is used as the bit string or flag to indicate the end of the frame.

1.2 Assumptions

- Input the bit
- make the input first charc in interger and store in diffrent variable
- after making that use xor and find the parity
- find the pattern for bitstuffing
- add the tailer in formate

1.3 Algorithm and Implementation

- Start
- Input the bit
- make the input first charc in interger and store in diffrent variable
- convert each string in the integer
- Finding Xor of the each corresponding bit string
- add the parity to the string
- Find the pattern with help of repplace of string and replace with particular pattern
- add the tailer to the string
- print the Transmitting string
- stop

1.4 Input and Output Format

- **Input Format**
  Enter binary bit data that has to be transmitted.

- **Output Formate**
  Print binary bit data with parity bit.
  Print the modified string that is to be transmitted

1.5 Flow chart

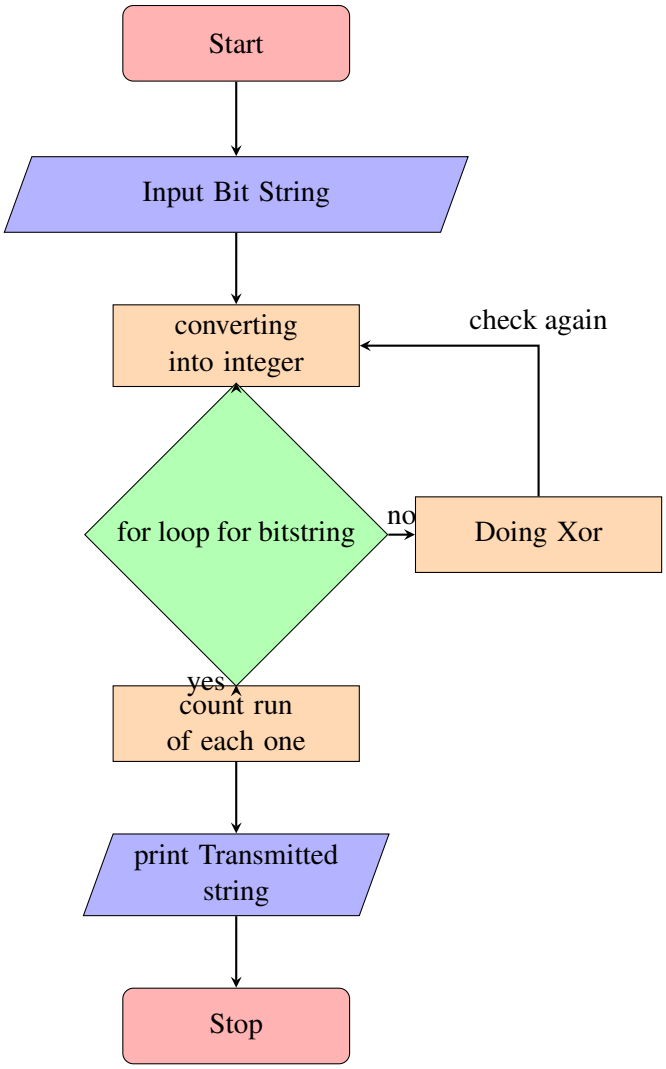

Figure 1: PS1

1.6 Screenshot

```
File  Edit  View  Search  Terminal  Help
satyam@admin17-OptiPlex-9020:~/Assignment_8$ python3 ps1.py
Enter the bit stream : 010101110100101
Parity bit data :0101011101001010
transmitting data :01001011101000100100101
satyam@admin17-OptiPlex-9020:~/Assignment_8$
```

Figure 2: Output

2 Problem Statement-2

2.1 Problem Statement

- **3X3 Numeric Tic-Tac-Toe**
  One player plays with the odd numbers (1, 3, 5, 7, 9) and the 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 starts 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 cells might be necessary to complete a different line.
  Note Line can be horizontal, vertical or diagonal

2.2 Algorithm and Implementation

- make a list of array
- define position of each array
- take input for every postion from the user
- find the value of and add for each user in the row coloum and digonal

2.3 Input and Output Format

- Print Welcome to the Game!
- Print whether it is Player 1s or Player 2s 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 flowchart

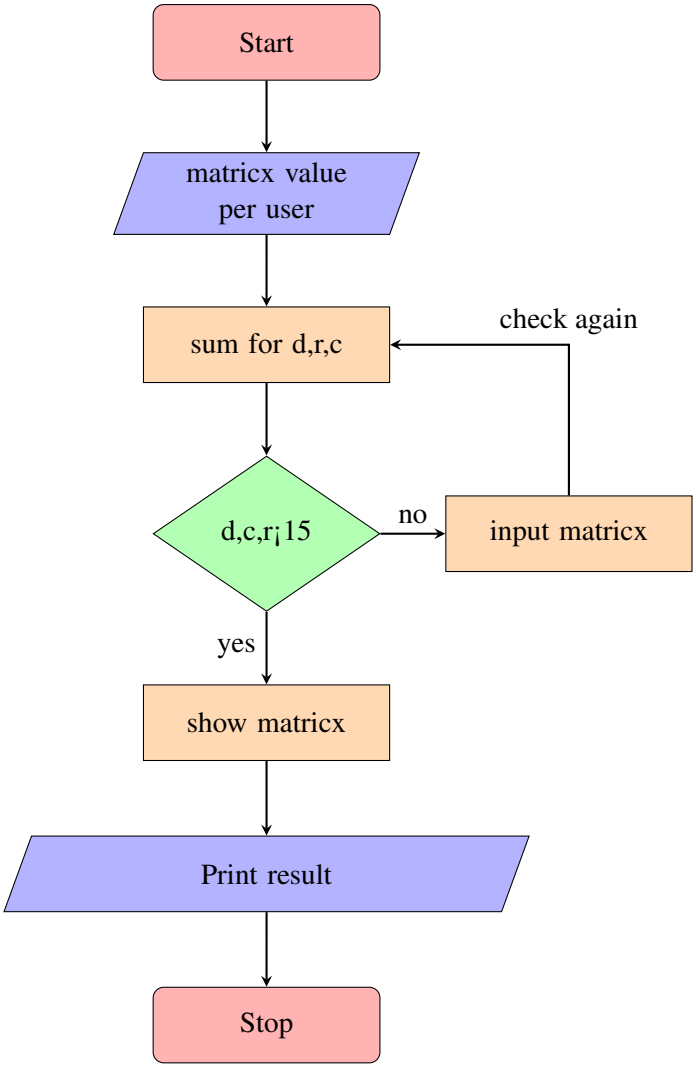
Start
matricx value
per user
sum for d,r,c
check again
d,c,r¡15
no
input matricx
yes
show matricx
Print result
Stop

Figure 3: PS2

2.5 Screenshot

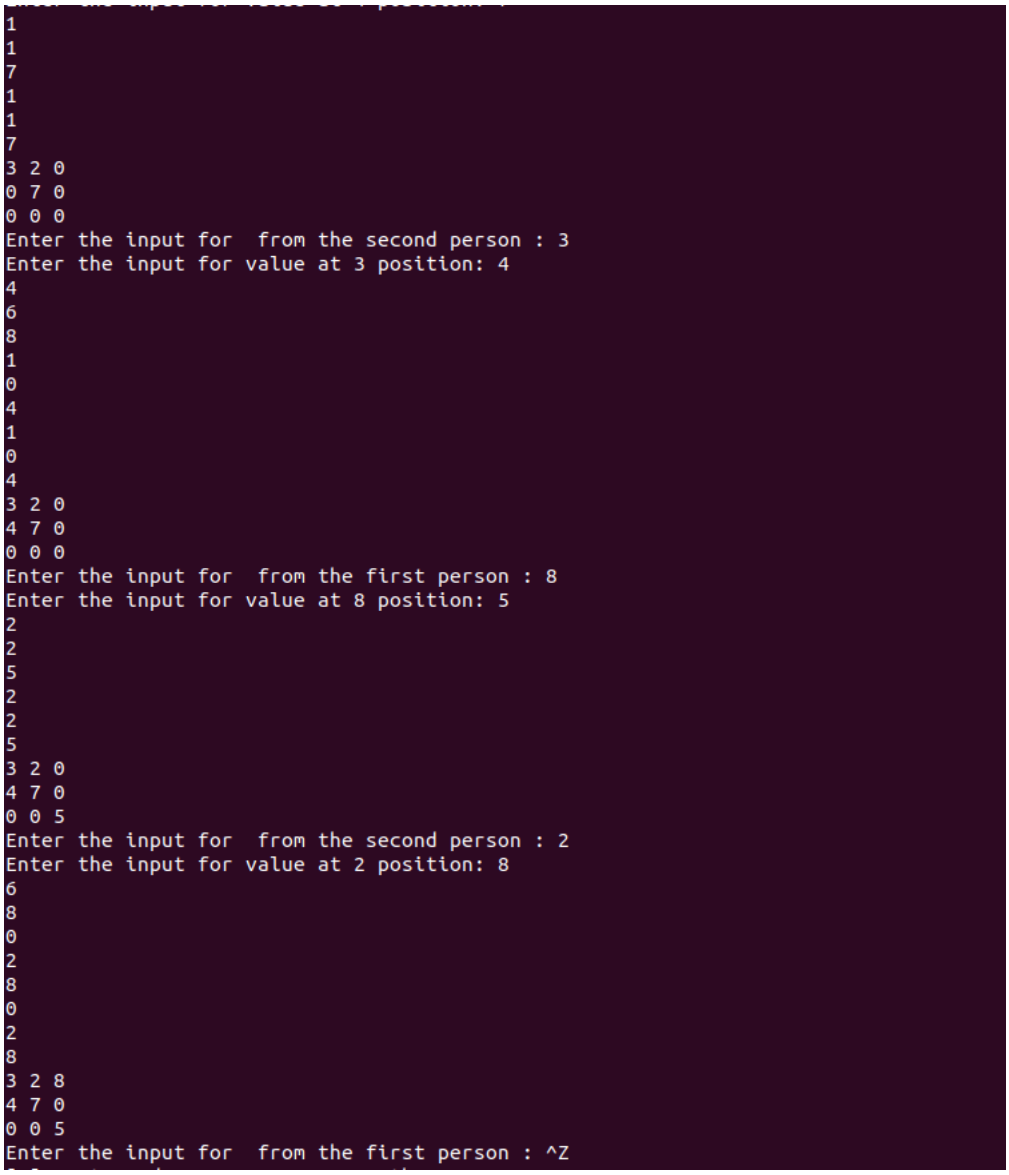

Figure 4: frequency of each alphabet

3 Appendix

3.1 Appendix-A : code for ps1

ps1.py

```
a = input("Enter the bit stream : ")
for char in a[0:1]:
    x = int(char)

for char in a[1:]:
    y = int(char)
    x ^= y

a += str(x)
print("Parity bit data :" + a)
a = a.replace('010', '0100')
a += '0101'
print("transmitting data :" + a)
```

3.2 Appendix-B : code for ps2

ps2.py

```
def value(x, y, arr):
    x = int(x)
    c = x // 3
    z = x % 3
    print(c)
    print(z)
    arr[c][z] = y
    print(arr[c][z])
    # For printing the matrix
    for i in range(0, 3):
        for j in range(0, 3):
            print(board[i][j], end=" ")
        print()


print("welcome to the game ")
board = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
for i in range(0, 3):
    for j in range(0, 3):
        print(board[i][j], end=" ")
    print()
fpers = ["1", "3", "5", "7"]
sepers = ["2", "4", "6", "8"]
for i in range(0, 9):
    if i % 2 == 0:
        a = input("Enter the input for  from the first person : ")
        if int(a) <= 9:
            d = 1
        else:
            print("Enter correct number.")
            exit()
        b = input("Enter the input for value at {} position: ".format(a))
        for char in fpers:
            if char == b:
                v = 1

            else:
                v = 0
        fpers.remove(b)
        #if v == 0:
           # print("Enter correct number.")
            #exit()
        a1 = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
        u = int(a)
        c = u // 3
        z = u % 3
        print(c)
        print(z)
        a1[c][z] = b
        print(a1[c][z])
        value(a, b, board)
        if (int(a1[0][0]) + int(a1[0][1]) + int(a1[0][2])) == 15:
            print("first win")
        if (int(a1[1][0]) + int(a1[1][1]) + int(a1[1][2])) == 15:
            print("first win")
```

```
        if (int(a1[2][0]) + int(a1[2][1]) + int(a1[2][2])) == 15:
            print("first win")
        if (int(a1[0][0]) + int(a1[1][0]) + int(a1[2][0])) == 15:
            print("first win")
        if (int(a1[0][1]) + int(a1[1][1]) + int(a1[2][1])) == 15:
            print("first win")
        if (int(a1[0][2]) + int(a1[1][2]) + int(a1[2][2])) == 15:
            print("first win")
        if (int(a1[0][0]) + int(a1[1][1]) + int(a1[2][2])) == 15:
            print("first win")
        if (int(a1[0][2]) + int(a1[1][1]) + int(a1[2][0])) == 15:
            print("first win")
```

```
    else:
        a = input("Enter the input for  from the second person : ")
        if int(a) <= 9:
            d = 1
        else:
            print("Enter correct number.")
            exit()
        b = input("Enter the input for value at {} position: ".format(a))
        for char in sepers:
            print(char)
            if char != b:
                v = 1
            else:
                v = 0
        sepers.remove(b)
        #if v == 1:
         #    print("Enter correct number.")
          #   exit()
        b1 = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
        u = int(a)
        c = u // 3
        z = u % 3
        print(c)
        print(z)
        b1[c][z] = b
        print(b1[c][z])
        value(a, b, board)

        if (int(b1[0][0]) + int(b1[0][1]) + int(b1[0][2])) == 15:
            print("first win")
        if (int(b1[1][0]) + int(b1[1][1]) + int(b1[1][2])) == 15:
            print("first win")
        if (int(b1[2][0]) + int(b1[2][1]) + int(b1[2][2])) == 15:
            print("first win")
        if (int(b1[0][0]) + int(b1[1][0]) + int(b1[2][0])) == 15:
            print("first win")
        if (int(b1[0][1]) + int(b1[1][1]) + int(b1[2][1])) == 15:
```

```
115             print("first win")
116         if (int(b1[0][2]) + int(b1[1][2]) + int(b1[2][2])) == 15:
117             print("first win")
118         if (int(b1[0][0]) + int(b1[1][1]) + int(b1[2][2])) ==15:
119             print("first win")
120         if (int(b1[0][2]) + int(b1[1][1]) + int(b1[2][0])) == 15:
121             print("first win")
```

References

[1] Become a git guru. https://www.atlassian.com/git/tutorials.

[2] geeksforgeeks for Python. https://www.geeksforgeeks.org/python-programming-language/.

[3] Git Hub repository. https://github.com/2019JTM2691/Assignment_8.

[4] Google python Class. https://www.youtube.com/watch?v=tKTZoB2Vjuk&list=PLC8825D0450647509.

[5] You tube Python. https://www.youtube.com/watch?v=rfscVS0vtbw&t=219s.