

Computer Networks-1 CMP405/CMPN405 Fall 2018

Lab2 –Framing with error detection

Objectives:

- Apply and simulate the "byte count" framing algorithm to input from user.
- Apply and simulate "even parity" error detection algorithm to frames.

Introduction:

In this lab you will write a program using Python or C++ "in **one file only** "that simulates two algorithms of the Data-link layer [framing with byte count and even parity error detection].

Example:

Input file:

PLAY

WORK

EAT

Output file:

```
[['00110110'], ['01010111'], ['01001111'], ['01010010'], ['01001011'], ['00110111']]
[['00110110'], ['01010000'], ['01001100'], ['01000001'], ['01011001'], ['00110010']]
[['00110101'], ['01000101'], ['01000001'], ['01010100'], ['01100101']]
```

Note that the binary representation of characters is just their ASCII code for more info about ASCII codes visit https://www.rapidtables.com/convert/number/hex-to-ascii.html

Program flow:

read input from file "each payload in a separate line"[1 mark]



convert each payload character to it's binary(ASCII) value[1 mark]



get the even parity of the payload characters and append it to the end of the frame[3 marks]



append the character count to the beginning of the frame and take into consideration the added [character count and the even parity bytes][1 mark]



print the frames into output files
[1], separate each frame and each
byte for readability[1 mark]



make a function that takes a frame as input (from file) and check the even parity and if it is correct it prints the original payload in characters, if it is not, it prints "error" to another output file.[3 marks]

Delivery:

Work in pairs and Send your <u>code file</u> and test files in a zip folder named with your names to salmacmpeg@gmail.com by 2/12/2018 midnight with subject [semester lab 2]