Assignment #A3P1
Computer Networks, Monsoon 2018. Mahavir Jhawar, Ashoka University
Submission Due: October 06, 2018

Marks for Sender side: 20
Marks for Receiver side: 20
Marks for team work: 10
Maximum marks a student can obtain: 30

In this assignment your team will be writing a program to implement CRC error detection technique. In particular your team must implement the following two component:

1. CRC Encoding (Sender Side)
2. CRC Decoding (Receiver Side)

Sender Side Program: CRC Encoding

1. The **Input** to your program is a text file F. Let m be a binary string representing the content of F. Split m into blocks of substrings. Each of these substrings m_i's must undergo the following process
   (a) Compute crc_{m_i} using CRC-32 polynomial (Given in the reference book). Let $m'_i = m_i \| \mathsf{crc}_{m_i}$.
   (b) Enclose m'_i into a frame using the program given in A1P1. The resulting frame must be of size no more than 500 bits.
2. Finally, all these frames are put together and the resulting binary string is denoted by F'.
3. **Output:** The F' is to be converted back into a text file titled: *f_send.txt*.

The file *f_send.txt* is not direclty given to the receiver's program. This file must be run through a medium that simulates a nosiy channel. For this, you are given the following program - **Medium.py**. All three programs, Medium.py, sender side and receiver side programs, must be placed in the same directory.

Instructions on how to introduce errors into *f_send.txt* using Medium.py are as follows:

1. Execute **Medium.py** with parameters: FLAG (the one used by your program); and p - the probability with which every bit gets flipped. For example, if FLAG used is 01111110 and $p = 0.1$, then run

   <CMD/Terminal> `python medium.py 01111110 0.1`

   Note that the FLAG section of each frame is exempt from the error.
2. **Output:** A text-file *f_err.txt* is generated after.

Receiver side program finally provided *f_err.txt* as input. Receiver side must implement CRC error detection procedure.

Receiver Side Program: CRC Error Detection

1. **Input:** *f_err.txt*.
2. First, the program must extract frames from the input file. For this you must use your receiver side program implemented as part of A1P1.
3. The program than checks every frame for error detection.
4. **Output:** A Text-file is to be generated containing each frame in its binary form (including start and end FLAGs) followed by a line containing: ***0*** - if no error is detected; ***1*** - if error gets detected. For example

```
01101111000011010 10
0
01101001011110101 01
1
01101001010101010 10
0
⋮
```

The output file is to be titled: *f_detect.txt*

Finally, execute

<CMD/Terminal> `python medium.py eval`

for measuring efficiency of the receiver side error detection program.