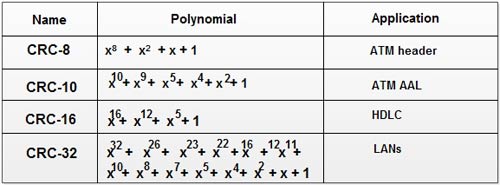
# Experiment 2: Error Detection using CRC

**Aim:** To apply CRC for error detection

**Objective:** After carrying out this experiment, students will be able to:

* Apply CRC to develop codes for error detection
* Analyze how this CRC is able to detect bit errors irrespective of their length and position in the data

**Problem statement:** You are required to write a program that uses CRC to detect burst errors in transmitted data. Initially, write the program using the CRC example you studied in class. Your final program should ask the user to input data and choose a generator polynomial from the list given in the figure below. Your program is required to calculate the checksum and the transmitted data. Subsequently, the user enters the received data. Applying the same generator polynomial on the received data should result in a remainder of 0.



**Analysis:** While analyzing your program, you are required to address the following points:

* How is this method different from 2D parity scheme that you have implemented previously?
* What are the limitations of this method of error detection?

**MARKS DISTRIBUTION**

|  |  |  |
| --- | --- | --- |
| **Component** | **Maximum Marks** | **Marks Obtained** |
| Algorithm/Flowchart | 7 |  |
| Program | 7 |  |
| Results/ Documentation | 6 |  |
| **Total** | **20** |  |

Submitted by: Harshit Kumar

Register No: 21ETMC412011

# Algorithm/Flowchart

• Start

• Enter the message to be transmitted

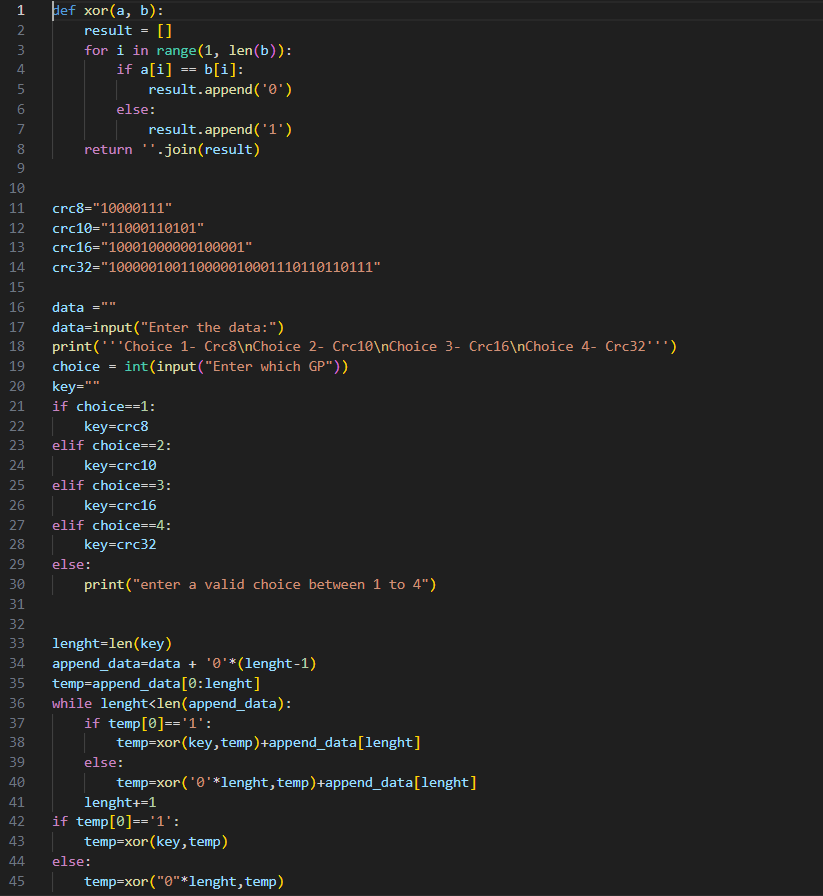
• Append the message with 16(since it is 16-bit CRC) 0`s (i.e. if you input 5 digit message, the appended message should be 21-bits.)

• XOR appended message and transmit it.(Here, you compare with an already exisitng string such as 10001000000100001 and replace the bits the same way XOR operation works)

• Verify the message that is received is the same as the one sent.

• End

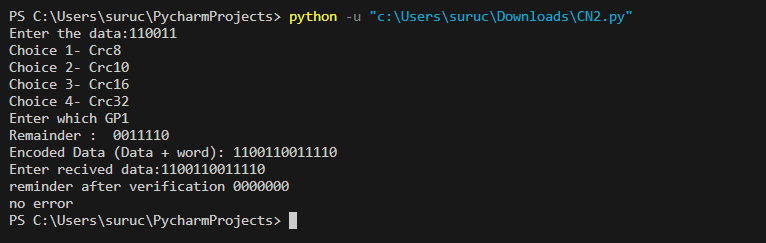
# Program





# Results

**Output:-**

****

# Analysis and Discussions

Analysis for 1st one

Cyclic Redundancy Check (CRC)

CRC is a redundancy error technique used to determine the error.

**Following are the steps used in CRC for error detection:**

* In CRC technique, a string of n 0s is appended to the data unit, and this n number is less than the number of bits in a predetermined number, known as division which is n+1 bits.
* Secondly, the newly extended data is divided by a divisor using a process is known as binary division. The remainder generated from this division is known as CRC remainder.
* Thirdly, the CRC remainder replaces the appended 0s at the end of the original data. This newly generated unit is sent to the receiver.
* The receiver receives the data followed by the CRC remainder. The receiver will treat this whole unit as a single unit, and it is divided by the same divisor that was used to find the CRC remainder.

If the resultant of this division is zero which means that it has no error, and the data is accepted.

If the resultant of this division is not zero which means that the data consists of an error. Therefore, the data is discarded.

Analysis for 2nd one

Though the Cyclic Redundancy Check look like an authentication mechanism, it is non trivial and easy to crack mechanism. **It is not suitable for security purpose**. Without the error correcting mechanism using CRC alone will be a useless thing.

# Conclusions

After carrying out this experiment, we were able to :

* Apply CRC to develop codes for error detection
* Analyse how this CRC is able to detect bit errors irrespective of their length and position in the data