Error Detection Techniques

General Instruction: Implement the following error checking methods using any programming language you like.

A. Simple Parity Check (Even Parity)

Description:

The program will simulate a simple parity check algorithm. Your program will accept 2 inputs *A* and *B*; *A* represents the original *k*-bit data word to send by the sender while *B* represents the *n*-bit codeword received in the receiver side. Assume that size of the data word is 8 bits.

Your program will consist of two parts - one for the sender and the other for the receiver. The part for the sender will be able to display the codeword out of the given data word. On the other hand, the part of the receiver should be able to check the syndrome *S* of the received codeword *B* and display the data word extracted from the codeword if the syndrome is 0, display "Discarded" otherwise.

Input

> Input A: 10010010
> Input B: 100000101

Output

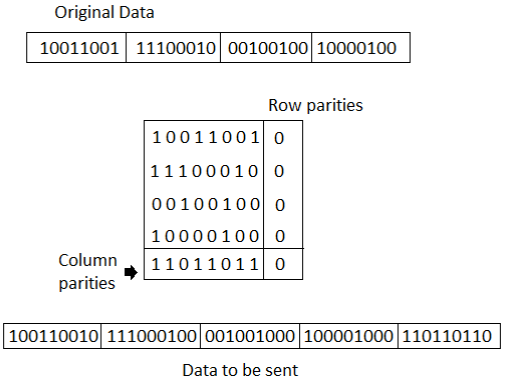
@Sender
Codeword: 100100101

@Receiver
Data word: Discarded

B. Two-dimensional Parity Check

Description:

The program will simulate a two-dimensional parity check algorithm at the receiver's side. Your program will accept a string of binary input that represents a stream of binary data whose size is 45 bits that consist four 8-bit data words, four 1-bit row parity bits and one 9-bit column parity bits as shown in an example below.



Your program will count how many bits in the four data words have changed along the transmission. Note, however, the limitations of two-dimensional parity check.

Input
> Data: 101100111 101010111 010110100 110101011 100101111

Output
Error count: 0

C. Checksum

Description:
The program will simulate checksum error detection method at the receiver's side. Your program will accept a string of binary input that represents a stream of binary data whose size is 40 bits that consist four 8-bit data words and 8-bit checksum. The program will display "Accept data" if no error is detected, otherwise display "Checksum error detected."

Input
> Data: 10011001 11100010 00100100 10000100 11011010

Output
Accept data

D. Cyclic Redundancy Check

Description:
The program will simulate cyclic redundancy check at the receiver's side. Your program will accept a string of binary input that represents a stream of binary data whose size is 7 bits that consist 4-bit data word and 3-bit CRC bits. The program will display "Accept data" if no error is detected, otherwise display "CRC error detected." Assume that the key word is 1011.

Input
> Data: 1001110

Output
Accept data