Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology, Pune-37

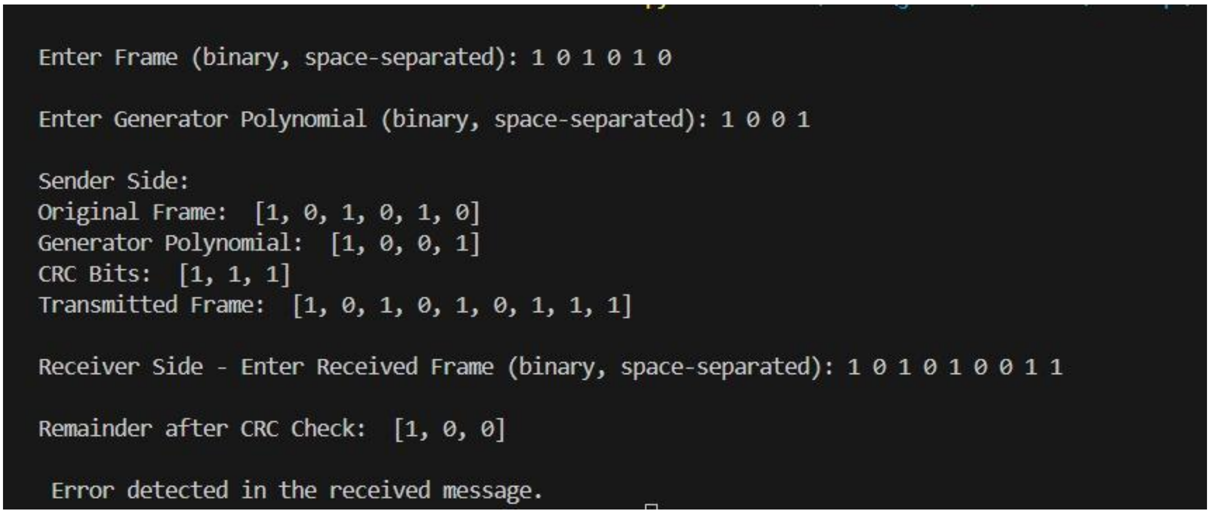
Department Of Artificial Intelligence and Data Science

COMPUTER NETWORK
Assignment 4

Class: - SY BTECH	Branch: - AIDS
Name : - Jineshwari Bagul	Year: -2024-25
Roll no - 26	

Write a program for error detection and correction for 7/8 bits ASCII codes using CRC.

```
def xor(dividend, divisor):
    """Perform XOR operation between dividend and divisor in binary division."""
    return [dividend[i] ^ divisor[i] for i in range(len(divisor))]

def crc_encode(frame, generator):
    """Encodes the frame by appending CRC bits using the generator polynomial."""
    fs, gs = len(frame), len(generator)
    extended_frame = frame + [0] * (gs - 1)  # Append zeros

    # Perform binary division
    temp = extended_frame[:]
    for i in range(fs):
        if temp[i] == 1:
            temp[i:i+gs] = xor(temp[i:i+gs], generator)

    crc_bits = temp[-(gs - 1):]  # Extract CRC
    transmitted_frame = frame + crc_bits
    return transmitted_frame, crc_bits

def crc_check(received_frame, generator):
    """Checks if received frame has errors using CRC verification."""
    fs, gs = len(received_frame) - (len(generator) - 1), len(generator)

    # Perform binary division
    temp = received_frame[:]
    for i in range(fs):
        if temp[i] == 1:
            temp[i:i+gs] = xor(temp[i:i+gs], generator)

    remainder = temp[-(gs - 1):]  # Extract remainder
    return remainder

def main():
    # Input Frame and Generator
    frame = list(map(int, input("\nEnter Frame (binary, space-separated): ").split()))
    generator = list(map(int, input("\nEnter Generator Polynomial (binary, space-separated): ").split()))

    # Encoding at Sender
    transmitted_frame, crc_bits = crc_encode(frame, generator)
    print("\nSender Side:")
    print("Original Frame: ", frame)
    print("Generator Polynomial: ", generator)
    print("CRC Bits: ", crc_bits)
    print("Transmitted Frame: ", transmitted_frame)

    # Receiver manually inputs the received frame
    received_frame = list(map(int, input("\nReceiver Side - Enter Received Frame (binary, space-separated): 
").split()))

    # Receiver CRC Check
    remainder = crc_check(received_frame, generator)
    print("\nRemainder after CRC Check: ", remainder)

    # Check for errors
    if all(bit == 0 for bit in remainder):
        print("\n Message received correctly, no errors detected.")
    else:
        print("\n Error detected in the received message.")

if _name_ == "_main_":
    main()
```



```
Enter Frame (binary, space-separated): 1 0 1 0 1 0

Enter Generator Polynomial (binary, space-separated): 1 0 0 1

Sender Side:
Original Frame:  [1, 0, 1, 0, 1, 0]
Generator Polynomial:  [1, 0, 0, 1]
CRC Bits:  [1, 1, 1]
Transmitted Frame:  [1, 0, 1, 0, 1, 0, 1, 1, 1]

Receiver Side - Enter Received Frame (binary, space-separated): 1 0 1 0 1 0 0 1 1

Remainder after CRC Check:  [1, 0, 0]

 Error detected in the received message.
```