NAME : Vinit Dhoot  
CLASS : SY-IT[A]

ROLL NO. : 45

SUBJECT : Computer Networks  
ASSIGNMENT NO. : 02

Problem Statement :Implement crc error detection method

CODE :

#include <stdio.h>

#include <string.h>

#define CRC\_Key\_LENGTH strlen(crcKey)

char inputData[28];

char crcResult[28];

char crcKey[10];

int inputDataLength, index\_i, index\_j;

void performCRC();

void XOR()

{

    printf("Remainder:-->%s\n", crcResult);

    for (index\_j = 1; index\_j < CRC\_Key\_LENGTH; index\_j++)

    {

        crcResult[index\_j] = ((crcResult[index\_j] == crcKey[index\_j]) ? '0' : '1');

    }

}

void receiverCheck()

{

    printf("\nData received: %s \n", inputData);

    performCRC();

    int errorDetected = 0;

    for (index\_i = 0; index\_i < CRC\_Key\_LENGTH - 1; index\_i++)

    {

        if (crcResult[index\_i] == '1')

        {

            errorDetected = 1;

            break;

        }

    }

    if (errorDetected)

    {

        printf("\nError in transmission \n\n");

    }

    else

    {

        printf("\nNo Error in transmission \n\n");

    }

}

void performCRC()

{

    for (index\_i = 0; index\_i < CRC\_Key\_LENGTH; index\_i++)

    {

        crcResult[index\_i] = inputData[index\_i];

    }

    do

    {

        if (crcResult[0] == '1')

        {

            XOR();

        }

        for (index\_j = 0; index\_j < CRC\_Key\_LENGTH - 1; index\_j++)

        {

            crcResult[index\_j] = crcResult[index\_j + 1];

        }

        crcResult[index\_j] = inputData[index\_i++];

    } while (index\_i <= inputDataLength + CRC\_Key\_LENGTH - 1);

    printf("%s  ", crcResult);

}

int main()

{

    printf("\nEnter data to be transmitted: ");

    scanf("%s", inputData);

    printf("\nEnter the CRC-Key: ");

    scanf("%s", crcKey);

    inputDataLength = strlen(inputData);

    for (index\_i = inputDataLength; index\_i < inputDataLength + CRC\_Key\_LENGTH - 1; index\_i++)

    {

        inputData[index\_i] = '0';

    }

    printf("\nData padded with n-1 zeros: %s \n", inputData);

    performCRC();

    printf("\nCRC Key Append value is: %s \n", crcResult);

    printf("\nChoose an option:\n");

    printf("1. Transmit correct data (No error)\n");

    printf("2. Introduce an error\n");

    int option;

    scanf("%d", &option);

    if (option == 1)

    {

        for (index\_i = inputDataLength; index\_i < inputDataLength + CRC\_Key\_LENGTH - 1; index\_i++)

        {

            inputData[index\_i] = crcResult[index\_i - inputDataLength];

        }

    }

    else if (option == 2)

    {

        printf("\nEnter the index of the bit to flip (0 to %lu): ", inputDataLength + CRC\_Key\_LENGTH - 2);

        int flipIndex;

        scanf("%d", &flipIndex);

        if (flipIndex >= 0 && flipIndex <= inputDataLength + CRC\_Key\_LENGTH - 2)

        {

            inputData[flipIndex] = (inputData[flipIndex] == '0') ? '1' : '0';

            printf("\nBit at index %d flipped.", flipIndex);

        }

        else

        {

            printf("\nInvalid index. No bit flipped.\n");

        }

    }

    printf("\nData at the sender's end: %s \n", inputData);

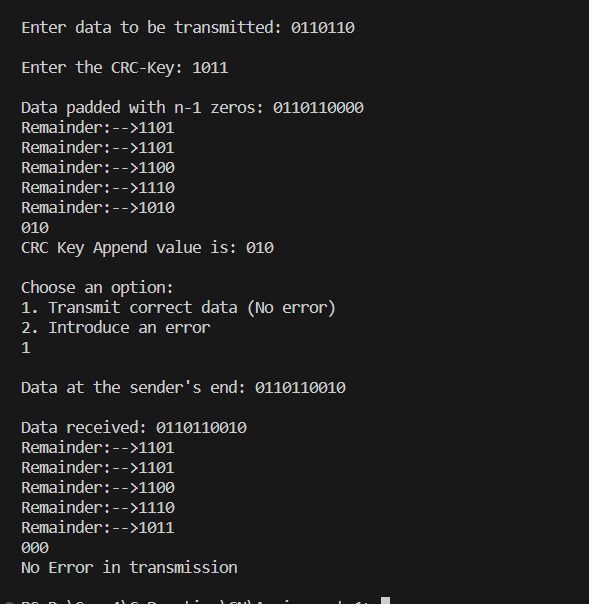
    receiverCheck();

    return 0;

}

OUTPUT :

Without error



With error :

