Ex No: 4 **Implement on a data set of characters the three CRC polynomials –CRC 12, CRC 16 and CRC CCIP.**

AIM:

To implement on a data set of characters the three CRC polynomials – CRC 12,CRC 16

and CRC CCIP.

ALGORITHM:

Step 1: Start the program.

Step 2: Add header file and data types to program.

Step 3: Add received data and checksum in print statement.

Step 4: Add else statement to check whether it is error or not.

Step 5: End the program

CODING:

```
// Include headers
#include<stdio.h>
#include<string.h>
// length of the generator polynomial
#define N strlen(gen_poly)
// data to be transmitted and received
char data[28];
// CRC value
char check_value[28];
// generator polynomial
char gen_poly[10];
// variables
int data_length,i,j;
// function that performs XOR operation
void XOR(){
  // if both bits are the same, the output is 0
  // if the bits are different the output is 1
```

```
    for(j = 1;j < N; j++)
    check_value[j] = (( check_value[j] == gen_poly[j])?'0':'1');

}
// Function to check for errors on the receiver side
void receiver(){
// get the received data
    printf("Enter the received data: ");
    scanf("%s", data);
    printf("\n-----------------------------\n");
    printf("Data received: %s", data);
// Cyclic Redundancy Check
    crc();
// Check if the remainder is zero to find the error
    for(i=0;(i<N-1) && (check_value[i]!='1');i++);
        if(i<N-1)
            printf("\nError detected\n\n");
        else
            printf("\nNo error detected\n\n");
}

void crc(){
    // initializing check_value
    for(i=0;i<N;i++)
        check_value[i]=data[i];
    do{
    // check if the first bit is 1 and calls XOR function
        if(check_value[0]=='1')
            XOR();
// Move the bits by 1 position for the next computation
        for(j=0;j<N-1;j++)
            check_value[j]=check_value[j+1];
        // appending a bit from data
        check_value[j]=data[i++];
    }while(i<=data_length+N-1);
// loop until the data ends
}
```

```
int main()
{
   // get the data to be transmitted
   printf("\nEnter data to be transmitted: ");
   scanf("%s",data);
   printf("\n Enter the Generating polynomial: ");
   // get the generator polynomial
   scanf("%s",gen_poly);
   // find the length of data
   data_length=strlen(data);
   // appending n-1 zeros to the data
   for(i=data_length;i<data_length+N-1;i++)
      data[i]='0';
   printf("\n---------------------------------------");
// print the data with padded zeros
   printf("\n Data padded with n-1 zeros : %s",data);
   printf("\n---------------------------------------");
// Cyclic Redundancy Check
   crc();
// print the computed check value
   printf("\nCRC or Check value is : %s",check_value);
// Append data with check_value(CRC)
   for(i=data_length;i<data_length+N-1;i++)
      data[i]=check_value[i-data_length];
   printf("\n---------------------------------------");
// printing the final data to be sent
   printf("\n Final data to be sent : %s",data);
   printf("\n---------------------------------------\n");
// Calling the receiver function to check errors
   receiver();
      return 0;
}
```

SAMPLE OUTPUT

```
Enter data to be transmitted: 1001101
Enter the Generating polynomial: 1011
---------------------------------------
```

```
Data padded with n-1 zeros : 1001101000
----------------------------------------
CRC or Check value is : 101
----------------------------------------
Final data to be sent : 1001101101
----------------------------------------
Enter the received data: 1001101101
----------------------------
Data received: 1001101101
No error detected
```