

CYCLE 2

AIM:

Write a program for error detecting code using CRC-CCITT (16-bits).

PROGRAM:

12-8-23

CYCLE-2

Write a program for error detecting code using CRC-CCITT (16-bits)

```
#include <stdio.h>
#include <string.h>
#define N strlen(divisor)
char data[30];
char rem[30];
char division[10];
int dlength, i, j;
void xor()
{
    for (j=1; j<N; j++)
        rem[j] = ((rem[j] == divisor[j]) ? '0' : '1');
}
void crc()
{
    for (i=0; i<N; i++)
        rem[i] = data[i];
    do {
        if (rem[0] == '1') xor();
        for (j=0; j<N-1; j++)
            rem[j] = rem[j+1];
            rem[j] = data[i+1];
    } while (i <= dlength + 16);
}
void receiver()
{
    printf("Enter the data being received: ");
    scanf("%s", data);
    printf("Data received: %s", data);
    crc();
    for (i=0; (i<N-1) && (rem[0] != '1'); i++)
        if (i<N-1) printf("\n Error detection\n in data\n");
}
```

८५८ (१);

CRC-16

1) Enter the data to be transmitted: 1001101
Enter the divisor: 1011
Data padded with ¹⁶ ~~more~~ zeros: 10011010000000000000
The remainder or CRC is: ~~101~~ 111
Final data being sent: 10011010000000000000111
Enter the data being received: 10011010000000000000111
Data received: 10011010000000000000111
No error detected in data.

2. Enter the data to be transmitted: 1001101

Enter the divisor: 1011

Data padded with ~~5~~ zeros: 1001101000000000

The remainder or CRC is: 101

Final data being sent: 1001101000000000101

Enter the data being received: 1001101000000000101

Data received: 100110100000000010111

Error detected in data.

$$\begin{array}{r} 1010 \\ \times 22 \\ \hline 220 \\ 2200 \\ \hline 2420 \end{array}$$

PROGRAM:

```
#include<stdio.h>
#include<string.h>
#define N strlen(divisor)
char data[28];
char rem[28];
char divisor[10];
int dlength,i,j;
void XOR(){
    for(j = 1;j < N; j++)
        rem[j] = (( rem[j] == divisor[j])?'0':'1');
}

void receiver(){

    printf("Enter the received data: ");
    scanf("%s", data);
    printf("\n\n");
    printf("Data received: %s", data);

    crc();

    for(i=0;(i<N-1) && (rem[i]!='1');i++);
    if(i<N-1)
        printf("\nError detected\n\n");
    else
        printf("\nNo error detected\n\n");
}

void crc(){

    for(i=0;i<N;i++)
        rem[i]=data[i];
    do{

        if(rem[0]=='1')
            XOR();

        for(j=0;j<N-1;j++)
            rem[j]=rem[j+1];
```

```

        rem[j]=data[i++];
    }
    while(i<=dlength+16);

}

int main()
{ int c=0;

    printf("\nEnter data to be transmitted: ");
    scanf("%s",data);
    printf("\n Enter the Divisor: ");
    scanf("%s",divisor);
    dlength=strlen(data);
    for(i=dlength;i<dlength+16;i++)
        data[i]='0';
    printf("\n");
    printf("\n Data padded with n-1 zeros : %s",data);
    printf("\n");
    crc();
    printf("\nCRC or Check value is : %s",rem);
    printf("\n rem strlen is : %d ", strlen(rem));
    for(i=dlength+13;i<dlength+16;i++)
    {   printf("\n %s",data);
        data[i]= rem[c++];
    }
    printf("\n");

    printf("\n Final data to be sent : %s",data);
    printf("\n\n");

    receiver();
    return 0;
}

```

Output:

```
Enter data to be transmitted: 1001101

Enter the Divisor: 1011

Data padded with n-1 zeros : 100110100000000000000000

CRC or Check value is : 111
rem strlen is : 3
100110100000000000000000
1001101000000000000000100
1001101000000000000000110

Final data to be sent : 100110100000000000000111

Enter the received data: 100110100000000000000111

Data received: 100110100000000000000111
No error detected
```

```
Enter data to be transmitted: 1001101

Enter the Divisor: 1011

Data padded with n-1 zeros : 100110100000000000000000

CRC or Check value is : 111
rem strlen is : 3
100110100000000000000000
1001101000000000000000100
1001101000000000000000110

Final data to be sent : 100110100000000000000111

Enter the received data: 100110100000000000000111

Data received: 100110100000000000000111
Error detected
```

```
...Program finished with exit code 0
Press ENTER to exit console. □
```