Experiment-5

Date: Implement CRC (CYCLIC REDUNDANCY CHECK)

AIM: To Implement Cyclic Redundancy Check

ALGORITHM: CRC works by adding the remainder to the end of data word (after dividing by a divisor) and sending it and again dividing the code word to check error. If there is no error, then remainder is zero. If remainder is not zero, then there is an error.

CODE:

```
#include<stdio.h>
#include<string.h>
int main()
{
int i,j,databit,databit,count=0;
char data[100],div[30],temp[30],quot[100],
rem[30],dvd[30];
printf("Enter no of bits\n");
scanf("%d",&databit);
printf("Enter data\n");
```

```
scanf("%s", data);
printf("Enter no of bits in divisor\n");
scanf("%d", &divbit);
printf("Enter divisor");
scanf("%s", div);
strcpy(divl, div);
for(i=0; i<divbit-1; i++)
    data[databit+i] = '0';

for(i=0; i<divbit; i++)
    temp[i] = data[i];

for(i=0; i<databit; i++)
{
    quot[i] = temp[0];
    if(quot[i]=='0')
        for(j=0; j<divbit; j++)
            div[j] = '0';
    else
        for(j=0; j<divbit; j++)
            div[j] = divl[j];
    for(j=divbit-1; j>0; j--)
    {
        if(temp[j]==div[j])
            rem[j-1] = '0';
```

```
        else
            rem[j-1] = '1';
    }
    rem[divbit-1] = data[i+divbit];
    strcpy(temp, rem);
}

strcpy(rem, temp);
printf("\n Remainder is:");
for(i=0; i<divbit-1; i++)
    printf("%c", rem[i]);
printf("\n CRC code is:");
for(i=0; i<divbit; i++)
    printf("%c", data[i]);
for(i=0; i<divbit-1; i++)
    printf("%c", rem[i]);
return 0;
}
```

o/p:

Enter the numbr of bits in data: 7

Enter data: 1110011

Enter no of bits in divisor: 3

Enter divisor: 101

Remainder is: 01

CRC code is: 1110011