Cycle-2

WAP for error detecting code using CRC-CCITT (16 bits)

```
#include<stdio.h>
#include<string.h>
#define N strlen(g)

char t[28], cs[28], g[10];
int a,e,c;
void xorfunction(){
 for(c=1;c<N;c++)
  cs[c]= ((cs[c] == g[c])?'0':'1');
}

void crc(){
  for(e=0;e<N;e++)
     cs[e]= t[e];

  do{
     if(cs[0]=='1')
       xorfunction();

     for(c=0;c<N-1;c++)
        cs[c]=cs[c+1];

     cs[c] = t[e++];
  }while(e<=a+N-1);
}

int main(){
  printf("\n Enter data: ");
  scanf("%s",t);
  printf("\n Enter generating polynomial: ");
```

```
#include <stdio.h>
#include <stdlib.h>
#include<stdio.h>
#include <conio.h>
#include <string.h>

void main(){
 int i,j,key,len,msglen;
 char input[100],key[30],temp[30],quot[100],rem[30],
 key1[30];

Printf("Enter data: ");
gets(input);
Printf("Enter G(x): ");
gets(key);
keylen = strlen(key);
 msglen = strlen(input);
 strcy(key1,key);
for(i=0;i<keylen-1;i++){
input[msglen+i]='0'
}
for(i=0;i<keylen;i++)
 temp[i]=input[i];
for(i=0;i<msglen;i++)
quot[i]=temp[0]
if(quot[i]=='0')
for(j=0;j<keylen;j++)
key[j]='0';
 else
for(j=0;j<keylen;j++)
key[j]=key1[j];
```

```
for ( j= keylen-1; j>0; j--) {
if (temp[j]== key[j])
rem[j-1] = '0'; else
 rem[j-1] = '1';

}
rem[keylen-1] = input[i+keylen];
strcpy(temp, rem);

}
strcpy (rem, temp);
printf("\n Quotient is ");
for (i=0; i<msglen ; i++)
printf ("%c", quot[i]);
printf ("\n Remainder is ");
for (i=0; i<keylen-1; i++)
printf("%c", rem[i]);
printf("\n Modified data is: ");
for (i=0; i<msglen; i++)
printf("%c", input[i]);
for (i=0; i<keylen-1; i++)
printf("%c", rem[i]);
  strcat (input, rem);

   keylen = strlen(key);
msglen = strlen(input);
strcpy (key1, key);
for (i=0; i<keylen-1; i++)
  input[msglen+i] = '0';

for (i=0; i<keylen; i++)
  temp[i] = input[i];

for (i=0; i<msglen; i++){
 quot[i] = temp[0];
```

```
if (quot[i] == '0')
for (j=0; j< keylen; j++)
key[j] = '0';
else
 for (j=0; j< keylen; j++)
 key[j] = key1[j];
 for (j= keylen-1; j>0; j--)
 if (temp[j-1] == key[j])
 rem[j-1] = '0'
 else
 rem[j-1] = '1';
}
rem[keylen-1] = input[i+keylen];
strcpy(temp, rem);
}
strcpy(rem, temp);
Printf("\n Remainder is at receiver side ");
for(i=0; i<keylen-1; i++)
Printf("%c", rem[i]);
getch();
}
```

O/P:

Case-i: Enter data: 1011010101
Enter g(x): 1010 → CRC-16 only $\frac{x^{16}+x^{12}+x^5+1}{\text{Fixed}}$ (DW, D)
Quotient: 1001000100
Remainder: 000
modified: 1011010101000 → CW
checking for error
Case-ii: Enter data: 1011010101000
G(x): 1010
Remainder: 000

CW = R
↳ CW = QP
↳ R = 0 ✓
R ≠ 0 Error detected

Case - 3) Enter input : 1011010101

Transmitted message - 1011010101111101110111010

Enter received message:

11111111100000000111111111

Error in data transmission has occured