Write a program for error detecting code using CRC-CCITT.

```
#include <stdio.h>
char m[50], g[50], r[50], q[50], temp[50];
void caltrans (int);
void calram (); → void crc(int);
void shiftl();

int main() {
    int n, i=0;
    char ch, flag=0;
    printf("enter the frame bits:");
    while ((ch= getc(stdin)) != '\n')
        m[i++]=ch;
        n=i;
        for (i=0; i<16; i++)
        m[n++]= '0';
        m[n]='\0';
printf(" message after appending 16 zeroes: %s",m);
  ~~for (i=0; i<=16; i++)~~
    ~~g[i]='0';~~
  g[0]=g[4]=g[11]=g[16]='1';
  g[17]='\0';
  printf("\n generator: %s",g);
  crc(n);
  printf("\n quotient: %s",q);
  caltrans(n);
  printf("\n transmitted frame: %s",m);
  printf("\n enter recieved frame:");
  scanf("\n %s",m);
  printf("CRC checking\n");
  crc (n);
 prinf ("\n last remainder : %s", r);
 for (i=0; i<16; i++)
   if (r[i]!='0')
     flag=1;
   else
     continue;
 if(flag==1)
 printf("Error during transmission');
 else
 prinf ( "\n Recieved frame is correct");
}
void crc (int n) {
   int i, j;
   for (i=0; i<n; i++)
   temp[i]=m[i];
  for (i=0; i<16; i++)
     r[i]=m[i]
  for (i=0; i<n-16; i++) {
     if (r[0]=='1') {
           q[i]='1';
           calram();
       } else {
           q[i]='0';
           shiftl();
       }
   r[16]=m[17+i];
     r[17]='\0';
     for (j=0; j<=17; j++)
       temp[j]=r[j];
     }
  q [n-16]='\0';
}
void calram() {
  ~~int calc~~ int i,j;
       for (i=1; i<=16; i++)
    r[i-1]=((int)temp[i]-48)^((int)
                        g[i]-48)+48;
    }
void shift() {
    int i;
    for (i=1; i<=16; i++)
      r[i-1]=r[i]; }
void caltrans(int n){
      int i, k=0;
      for (i=n-16; i<n; i++)
   m[i]=((int)m[i]-48)^((int)r[k++]-
                              48)+48;
      m[i]='\0'; ~~8~~ return 0; }
```

OUTPUT-

```
enter the frame bits: 1011
message after appending 16 zeroes:
                  1011000000000000
generator: 10001000000100001
quotient: 1011
transmitted frame: 10111011000101101011
crc checking
last mechanism: 0000000000000000
recieved frame is correct.
```