

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

WAP 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 crc(int);
void calxam();
void shift();

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 zeros: %s", m);
    for (i = 0; i <= 16; i++)
        g[i] = '0';
    g[0] = g[4] = g[17] = g[18] = '1'; g[17] = '10';
    printf("In generator: %s", g);
    printf("In quotient: %s", q);
    caltrans(n);
    printf("In transmitted frame: %s", m);
    printf("Enter transmitted frame:");
    scanf("%s", m);
```

```

Printf ("CRC checking\n");
crc (n);
Printf ("In n last remainder: %s", r);
for (i=0; i<16; i++)
    if (r[i] != '0')
        flag = 1;
    else
        continue;
    if (flag == 0)
        Printf ("Error during transmission");
    else
        Printf ("In n Received frame is correct");
}

```

```

void (ccrcint 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';
            (alxam);
        }
        else
        {
            q[i] = '0';
            shift (i);
        }
    }
}

```

```
    x[i] = m[i] + 1;
    x[i] = '0';
    for (j = 0; j < i; j++)
        x[j] = x[j];
}
```

```
    x[n-1] = '0';
}
```

```
void calsumc()
{
    int i;
```

```
    for (i = 1; i <= 16; i++)
```

```
        x[i-1] = (x[i] + 1) % 10;
```

```
void shift1c()
```

```
{
```

```
    int i;
```

```
    for (i = 1; i <= 16; i++)
```

```
        x[i-1] = x[i];
}
```

```
void caltrans(int n)
```

```
{
```

```
    int i, k = 0;
```

```
    for (i = n-1; i >= 0; i++)
```

```
        m[i] = (x[i] + 1) % 10;
```

```
        m[i] = '0';
}
```


Q18:-

Enter frame bits: 1011

message after appending 16 zeros:

10110000000000000000

generator: 1000100000100001

quotient: 1011

transmitted frame: 1011011000101101011

enter transmitted frame: 1011011000101101011

last remainder: 0000000000000000

Received frame is correct.

Write the algorithm
to verify it

Output:

```
Enter the frame bits:1001
Message after appending 16 zeros:10010000000000000000
Generator polynomial:10001000000100001

Transmitted frame:10010000000000000000
Enter the received frame:10010000000000000000
CRC Checking...

Last remainder:1001000100101001
Error during transmission
```