NAME - KAUSHAL OZA
ROLL NO - 40 (A)
SRN - 201900754
CHIT NO - 11

-- CN ESE LAB EXAMINATION --

Problem Statement - Program for calculating CRC using Modulo-2 method.

----

1) Algorithm -

CRC or Cyclic Redundancy Check is a method of detecting accidental changes/errors in the communication channel.

CRC uses Generator Polynomial which is available on both sender and receiver sides. An example generator polynomial is of the form of $x^3 + x + 1$. This generator polynomial represents key 1011. Another example is $x^2 + 1$ that represents key 101.

M is the message in bits.

N is the number of bits in data to be sent from the sender's side.

K is the number of bits in the key obtained from the generator's polynomial.

P is the generator polynomial or divisor.

MODULO - 2

The process of modulo-2 binary division is the same as the familiar division process we use for decimal numbers. Just that instead of subtraction, we use XOR here.

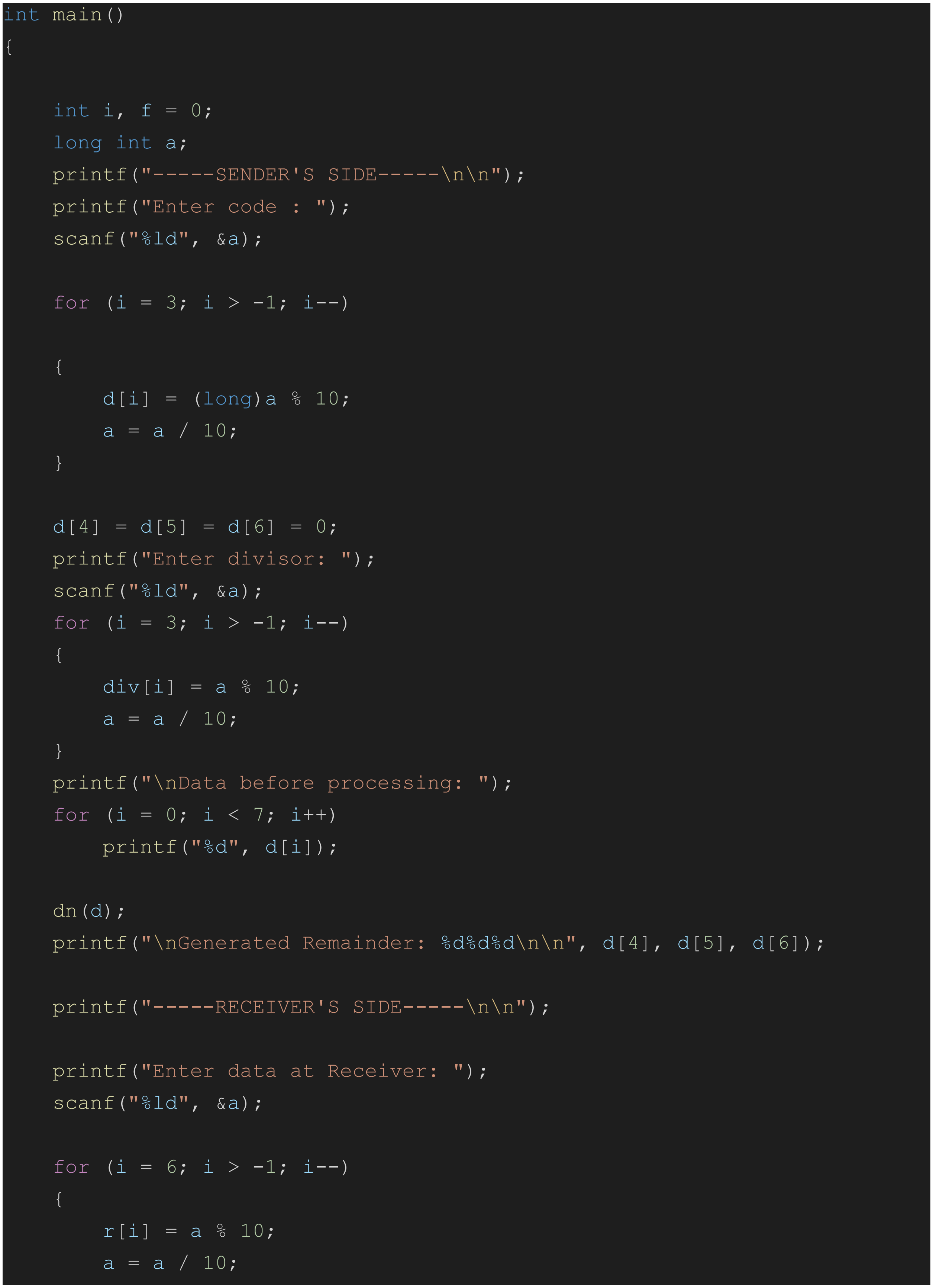
a) In each step, a copy of the divisor (or data) is XORed with the k bits of the dividend (or key).
b) The result of the XOR operation (remainder) is (n-1) bits, which is used for the next step after 1 extra bit is pulled down to make it n bits long.
c) When there are no bits left to pull down, we have a result. The (n-1)-bit remainder is appended at the sender side.
d) For example, if the data word to be sent is 100100 and the key is 1101. So, on the sender side the remainder we will be getting is 001 and the encoded data will be 100100001 (remainder appended to the data).
e) On the receiver's end, the code-word received will be 100100001. And therefore, the remainders should be all zeros to know that there was no error in the transmission.
f) Dividend=M+k-1 zeros and the Divisor will be the key.

FOR ERROR IN TRANSMISSION

g) For Error in transmission media, the Codeword received at the receiver side - 100000001. After solving this we won't be getting all the zeros which indicate that the error is detected at the receiver's side.

2) Code -

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

int d[7], div[4], r[7];
void dn(int dd[]);
```

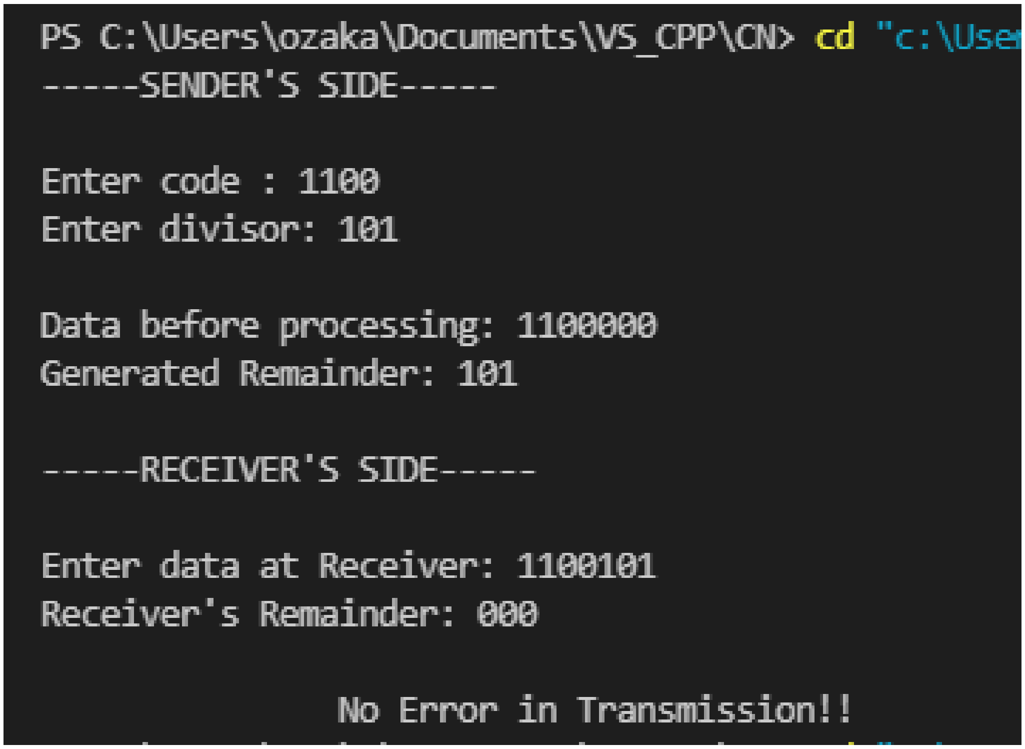


```
int main()
{

    int i, f = 0;
    long int a;
    printf("-----SENDER'S SIDE-----\n\n");
    printf("Enter code : ");
    scanf("%ld", &a);

    for (i = 3; i > -1; i--)

    {
        d[i] = (long)a % 10;
        a = a / 10;
    }

    d[4] = d[5] = d[6] = 0;
    printf("Enter divisor: ");
    scanf("%ld", &a);
    for (i = 3; i > -1; i--)
    {
        div[i] = a % 10;
        a = a / 10;
    }
    printf("\nData before processing: ");
    for (i = 0; i < 7; i++)
        printf("%d", d[i]);

    dn(d);
    printf("\nGenerated Remainder: %d%d%d\n\n", d[4], d[5], d[6]);

    printf("-----RECEIVER'S SIDE-----\n\n");

    printf("Enter data at Receiver: ");
    scanf("%ld", &a);

    for (i = 6; i > -1; i--)
    {
        r[i] = a % 10;
        a = a / 10;
```

```c
    }
    dn(r);

    for (i = 4; i < 7; i++)
        if (r[i] == 1)
        {
            f = 1;
            break;
        }
    printf("Receiver's Remainder: ");

    for (i = 4; i < 7; i++)
        printf("%d", r[i]);

    if (f == 1)
        printf("\n\n               Error in Transmission!!\n");
    else
        printf("\n\n               No Error in Transmission!!\n");
    return 0;
}

void dn(int dd[])
{
    int i, j;
    for (i = 0; i < 4; i++)
    {
        if (dd[i] == 0)
            continue;
        for (j = i + 3; j >= i; j--)
        {
            dd[j] = dd[j] ^ div[j - i];
        }
    }
}
```

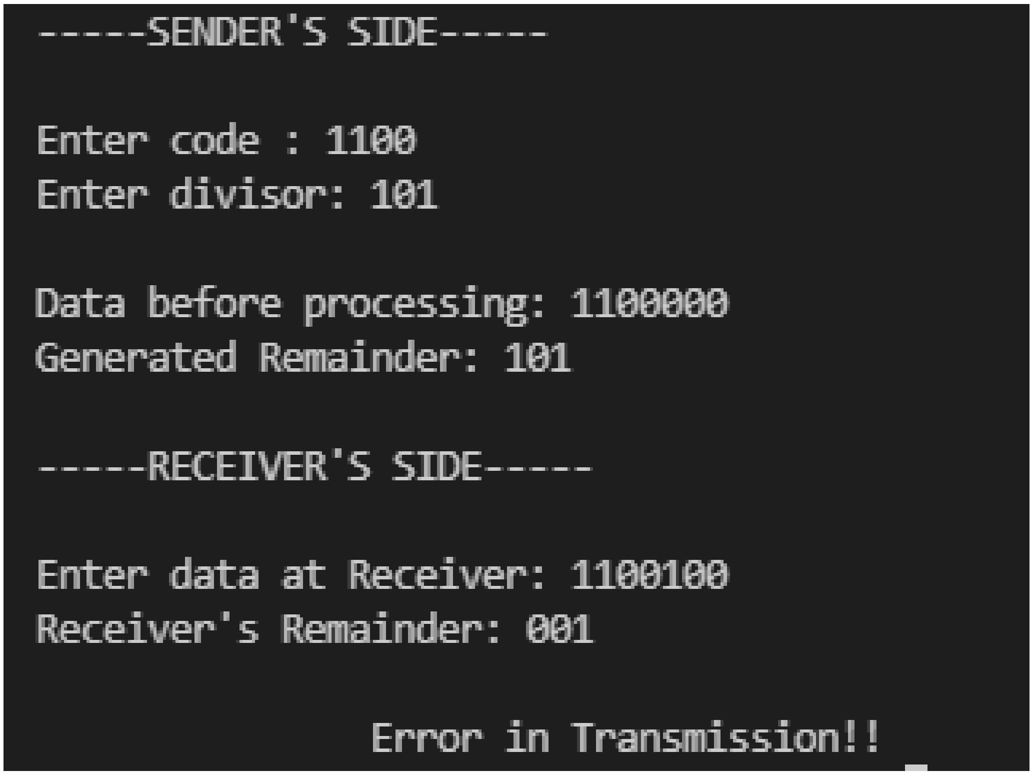
3) Screenshots/ Output -

a) TRANSMISSION WITHOUT ERROR -



```
PS C:\Users\ozaka\Documents\VS_CPP\CN> cd "c:\Use
-----SENDER'S SIDE-----

Enter code : 1100
Enter divisor: 101

Data before processing: 1100000
Generated Remainder: 101

-----RECEIVER'S SIDE-----

Enter data at Receiver: 1100101
Receiver's Remainder: 000

               No Error in Transmission!!
```

b) TRANSMISSION WITH ERROR -



```
-----SENDER'S SIDE-----

Enter code : 1100
Enter divisor: 101

Data before processing: 1100000
Generated Remainder: 101

-----RECEIVER'S SIDE-----

Enter data at Receiver: 1100100
Receiver's Remainder: 001

               Error in Transmission!!
```

--------40-----------40---------40-----------40----------40----------40------------