LAB 13

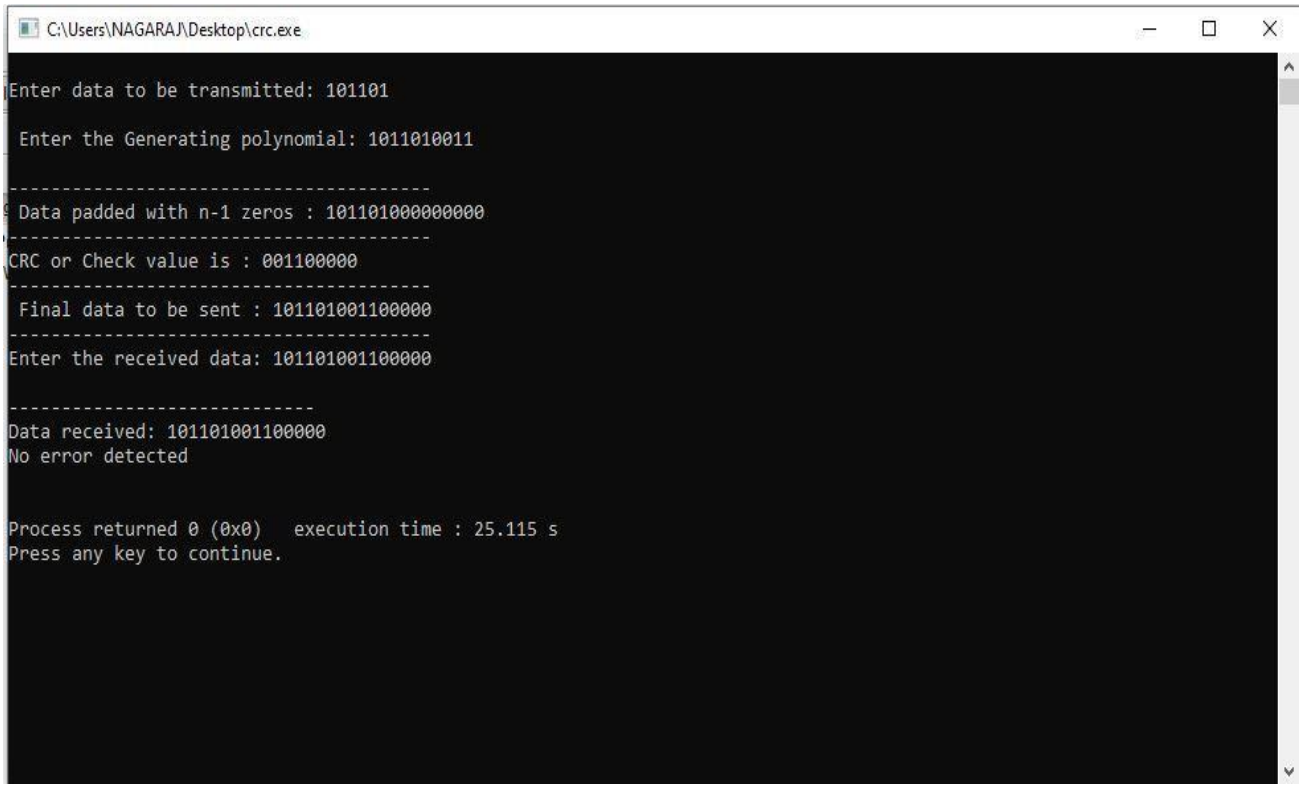
Write a program for error detecting code using CRCCCITT (16-bits).

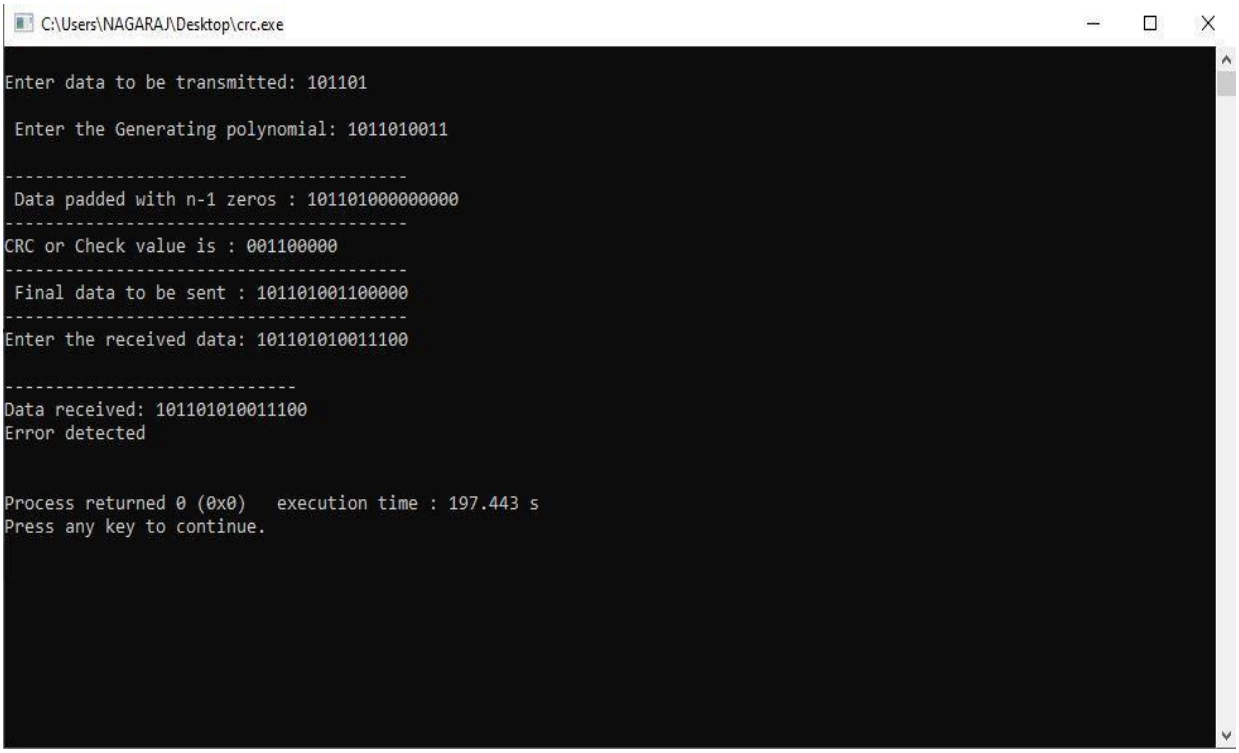
Code:

```
#include<stdio.h>
#include<string.h>
#define N strlen(gen_poly)
char data[28];
char check_value[28];
char  gen_poly[10];
int data_length,i,j;
void XOR(){
    for(j = 1;j < N; j++)
    check_value[j] = (( check_value[j] == gen_poly[j])?'0':'1');
}
void receiver(){
    printf("Enter the received data: ");
    scanf("%s", data);
    printf("\n.....\n");
    printf("Data received: %s", data);
    crc();
    for(i=0;(i<N-1) && (check_value[i]!='1');i++);
        if(i<N-1)
            printf("\nError detected\n\n");
        else
            printf("\nNo error detected\n\n");
}

void crc(){
    for(i=0;i<N;i++)
check_value[i]=data[i];
```

```
    do{
        if(check_value[0]=='1')
            XOR();
        for(j=0;j<N-1;j++)
            check_value[j]=check_value[j+1];
        check_value[j]=data[i++];
    }while(i<=data_length+N-1);
}

int main()
{
    printf("\nEnter data to be transmitted: ");
    scanf("%s",data);
    printf("\n Enter the Generating polynomial: ");
    scanf("%s",gen_poly);
    data_length=strlen(data);
    for(i=data_length;i<data_length+N-1;i++)
        data[i]='0';
    printf("\n----------");
    printf("\n Data padded with n-1 zeros : %s",data);
    printf("\n----------");
    crc();
    printf("\nCRC or Check value is : %s",check_value);
    for(i=data_length;i<data_length+N-1;i++)
        data[i]=check_value[i-data_length];
    printf("\n----------");
    printf("\n Final data to be sent : %s",data);
    printf("\n----------\n");
    receiver();
        return 0;
}
```

OUTPUT :


C:\Users\NAGARAJ\Desktop\crc.exe
Enter data to be transmitted: 101101
Enter the Generating polynomial: 1011010011
Data padded with n-1 zeros : 101101000000000
CRC or Check value is : 001100000
Final data to be sent : 101101001100000
Enter the received data: 101101001100000
Data received: 101101001100000
No error detected
Process returned 0 (0x0) execution time : 25.115 s
Press any key to continue.

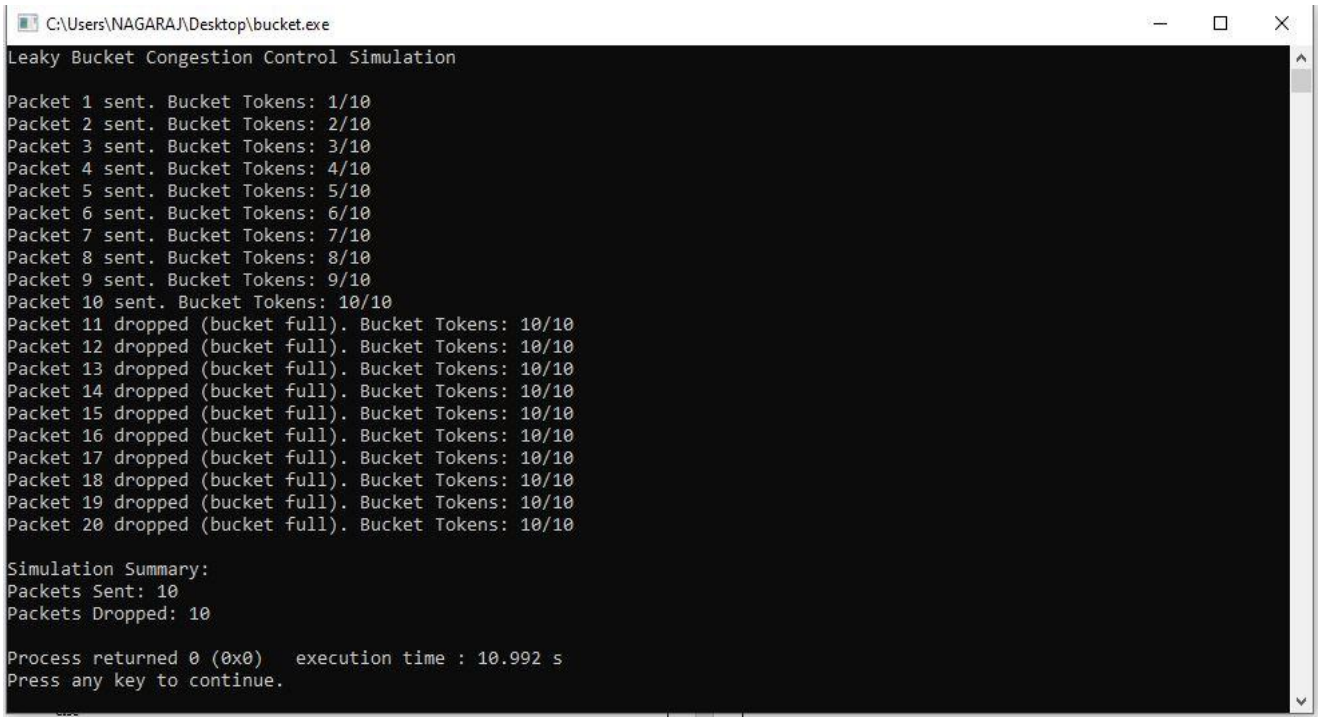

C:\Users\NAGARAJ\Desktop\crc.exe
Enter data to be transmitted: 101101
Enter the Generating polynomial: 1011010011
Data padded with n-1 zeros : 101101000000000
CRC or Check value is : 001100000
Final data to be sent : 101101001100000
Enter the received data: 101101010011100
Data received: 101101010011100
Error detected
Process returned 0 (0x0) execution time : 197.443 s
Press any key to continue.

b. Write a program for congestion control using Leaky bucket algorithm.

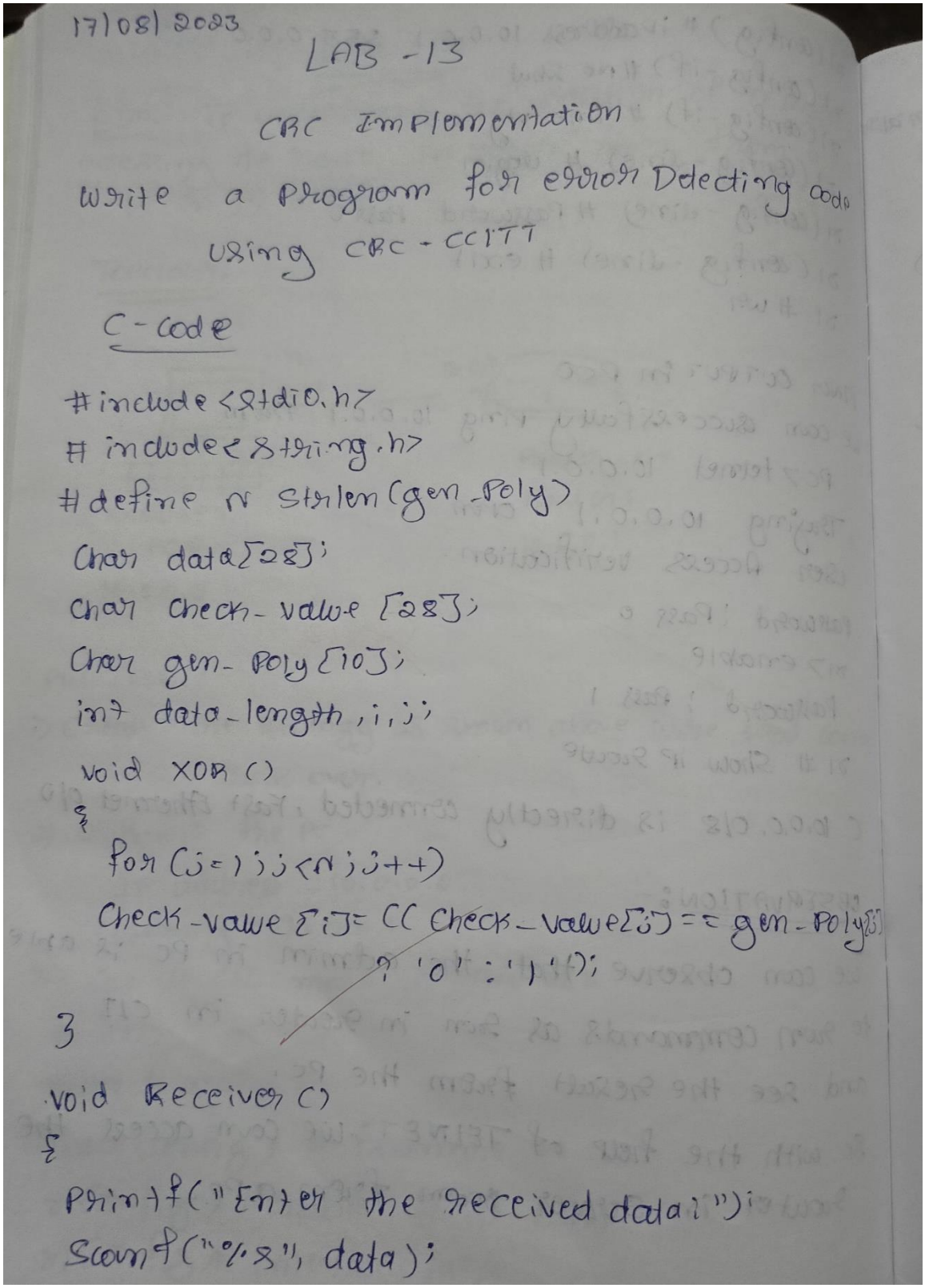
Code :

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#define BUCKET_SIZE 10
#define RATE 1
#define PACKETS 20

int main() {
    int bucket = 0;
    int sent = 0;
    int dropped = 0;
    int i;

    printf("Leaky Bucket Congestion Control Simulation\n\n");

    for (i = 0; i < PACKETS; i++) {
        usleep(500000);

        if (bucket < BUCKET_SIZE) {
            printf("Packet %d sent. Bucket Tokens: %d/%d\n", i + 1, bucket + 1,
BUCKET_SIZE);
            bucket++;
            sent++;
        } else {
            printf("Packet %d dropped (bucket full). Bucket Tokens: %d/%d\n", i + 1,
bucket, BUCKET_SIZE);
            dropped++;
        }
    }

    printf("\nSimulation Summary:\n");
    printf("Packets Sent: %d\n", sent);
    printf("Packets Dropped: %d\n", dropped);

    return 0;
}
```

Output :


C:\Users\NAGARAJ\Desktop\bucket.exe
Leaky Bucket Congestion Control Simulation
Packet 1 sent. Bucket Tokens: 1/10
Packet 2 sent. Bucket Tokens: 2/10
Packet 3 sent. Bucket Tokens: 3/10
Packet 4 sent. Bucket Tokens: 4/10
Packet 5 sent. Bucket Tokens: 5/10
Packet 6 sent. Bucket Tokens: 6/10
Packet 7 sent. Bucket Tokens: 7/10
Packet 8 sent. Bucket Tokens: 8/10
Packet 9 sent. Bucket Tokens: 9/10
Packet 10 sent. Bucket Tokens: 10/10
Packet 11 dropped (bucket full). Bucket Tokens: 10/10
Packet 12 dropped (bucket full). Bucket Tokens: 10/10
Packet 13 dropped (bucket full). Bucket Tokens: 10/10
Packet 14 dropped (bucket full). Bucket Tokens: 10/10
Packet 15 dropped (bucket full). Bucket Tokens: 10/10
Packet 16 dropped (bucket full). Bucket Tokens: 10/10
Packet 17 dropped (bucket full). Bucket Tokens: 10/10
Packet 18 dropped (bucket full). Bucket Tokens: 10/10
Packet 19 dropped (bucket full). Bucket Tokens: 10/10
Packet 20 dropped (bucket full). Bucket Tokens: 10/10
Simulation Summary:
Packets Sent: 10
Packets Dropped: 10
Process returned 0 (0x0) execution time : 10.992 s
Press any key to continue.

OBSERVATION :



17/08/2023

LAB - 13

CRC Implementation

Write a Program for error Detecting code using CRC - CCITT

C - code

```
#include<stdio.h>
#include<string.h>
#define N strlen(gen_poly)
char data[28];
char check_value[28];
char gen_poly[10];
int data_length,i,j;
void XOR()
{
  for(j=1;j<N;j++)
  check_value[j]=(check_value[j]==gen_poly[j])
                 ?'0':'1');
}
void Receiver()
{
printf("Enter the received data:");
scanf("%s",data);
```

```
printf("\n----------\n");
printf("Data received : %s", data);
crc();
for(i=0;(i<N-1) && (check_value[i]!='1');i++)
;
if(i<N-1)
printf("\n Error detected \n\n");
else
printf("\n No Error detected \n\n");
}
void crc()
{
for(i=0;i<N;i++)
check_value[i]=data[i];
do{
if(check_value[0]=='1')
XOR();
for(j=0;j<N-1;j++)
check_value[j]=data[i++];
}while(i<=data_length+N-1);
}
int main()
{
printf("\n Enter the Data to be
transmitted :");
```

```
Scanf("%s", data);
Printf("\n Enter the generating Polynomial : ");
Scanf("%s", gen_poly);
data_length = strlen(data);
for(i=data_length; i<data_length+N-1; i++)
    data[i]='0';
Printf("\n ----------");
Printf("\n Data Padded with n-1 zeros : %s",
        data);
Printf("\n ----------");
crc();
Printf("\n CRC or Check value is : %s",
        check_value);
for(i=data_length; i<data_length+N-1; i++)
    data[i] = check_value[i-data_length];
Printf("\n ----------");
Printf("\n Final data to be sent : %s", data);
Printf("\n ----------\n");
receiver();
return 0;
}
```

OUTPUT:

Enter data to be transmitted : 101010
Enter the divisor Polynomial : 1011
Data Padded with n-1 zeroes : 101010000
CRC value is : 001
Final codeword to be sent = 101010001

Enter the received data : 10001000
Error detected

Enter data to be transmitted : 101100
Enter the divisor Polynomial : 1001
Data Padded with n-1 zeroes : 101100000
CRC value is : 001
Final codeword to be sent : 101100001

Enter the received data : 101100001

No Error Detected.

18/8

Write a program for congestion control using Leaky Bucket algorithm

C- code

```
#include<stdio.h>
int main()
{
   int incoming, outgoing, buck_size, n, store
Printf("Enter Bucket size:");
Scanf("%d", &buck_size);
Printf ("Enter outgoing size:");
Scanf("%d", &outgoing);
Printf("Enter the number of inputs:")
Scanf("%d", &n);
while (n!=0)
{  Printf("Enter the incoming bucket size:")
   Scanf("%d", &incoming);
   if (incoming <= (buck_size - store))
   {
     store += incoming;
    Printf("Bucket buffer size %d out of
          %d' \n", store, buck_size)

  }
  else
  {
```

```
        incoming -(buck_size-store));
printf("Bucket buffer size %d out of %d \n",
        store, buck_size);
    store = store - outgoing
printf("After outgoing %d Packets left out of
    %d in buffer \n", store, buck_size);
    n--;
  }
}
```

18/8

OUTPUT:-

Enter bucket size: 5000
Enter outgoing rate: 2000
Enter number of inputs: 2
Enter the incoming Packet size: 3000
Bucket Buffer size 3000 out of 5000
After outgoing 1000 Packets left out of 5000
in buffer.
Enter the incoming Packet size: 1000
Bucket buffer size 2000 out of 5000
After outgoing 0 Packets left out of 5000
in Buffer.