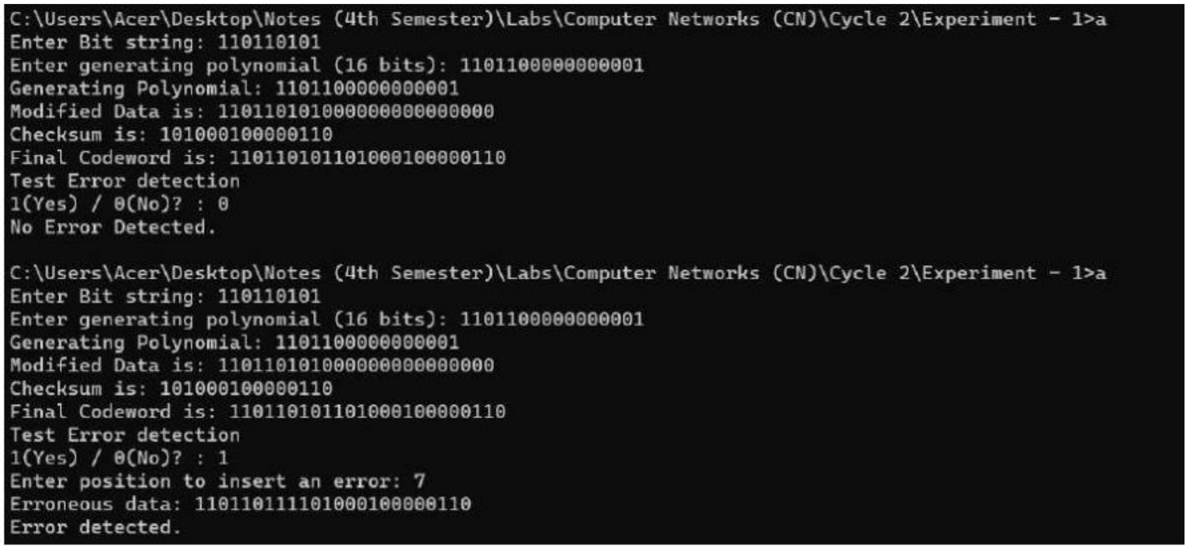
Lab 13

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

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

char data[100], crc[16], gen[17];
int len, i, j;

void calc_crc() {
    for (i = 0; i < strlen(gen); i++)
        crc[i] = data[i];
    do {
        if (crc[0] == '1') {
            for (j = 1; j < strlen(gen); j++)
                crc[j] = ((crc[j] == gen[j]) ? '0' : '1');
        }
        for (j = 0; j < strlen(gen) - 1; j++)
            crc[j] = crc[j + 1];
        crc[j] = data[i++];
    } while (i <= len + strlen(gen) - 1);
}

int main() {
    printf("Enter Bit string: ");
    scanf("%s", data);
    len = strlen(data);

    printf("Enter generating polynomial (16 bits): ");
    scanf("%s", gen);
    if (strlen(gen) != 16) {
        printf("Generator polynomial must be 16 bits.\n");
        return 1;
    }

    printf("Generating Polynomial: %s\n", gen);

    for (i = len; i < len + strlen(gen) - 1; i++)
        data[i] = '0';
    printf("Modified Data is: %s\n", data);
    calc_crc();
    printf("Checksum is: %s\n", crc);

    for (i = len; i < len + strlen(gen) - 1; i++)
        data[i] = crc[i - len];

    printf("Final Codeword is: %s\n", data);
    printf("Test Error detection\n1(Yes) / 0(No)? : ");
    scanf("%d", &i);

    if (i == 1) {
        printf("Enter position to insert an error: ");
        scanf("%d", &i);
        data[i] = (data[i] == '0') ? '1' : '0';
        printf("Erroneous data: %s\n", data);
    }

    calc_crc();

    for (i = 0; (i < strlen(gen) - 1) && (crc[i] != '1'); i++);

    if (i < strlen(gen) - 1)
        printf("Error detected.\n");
    else
        printf("No Error Detected.\n");

    return 0;
}
```



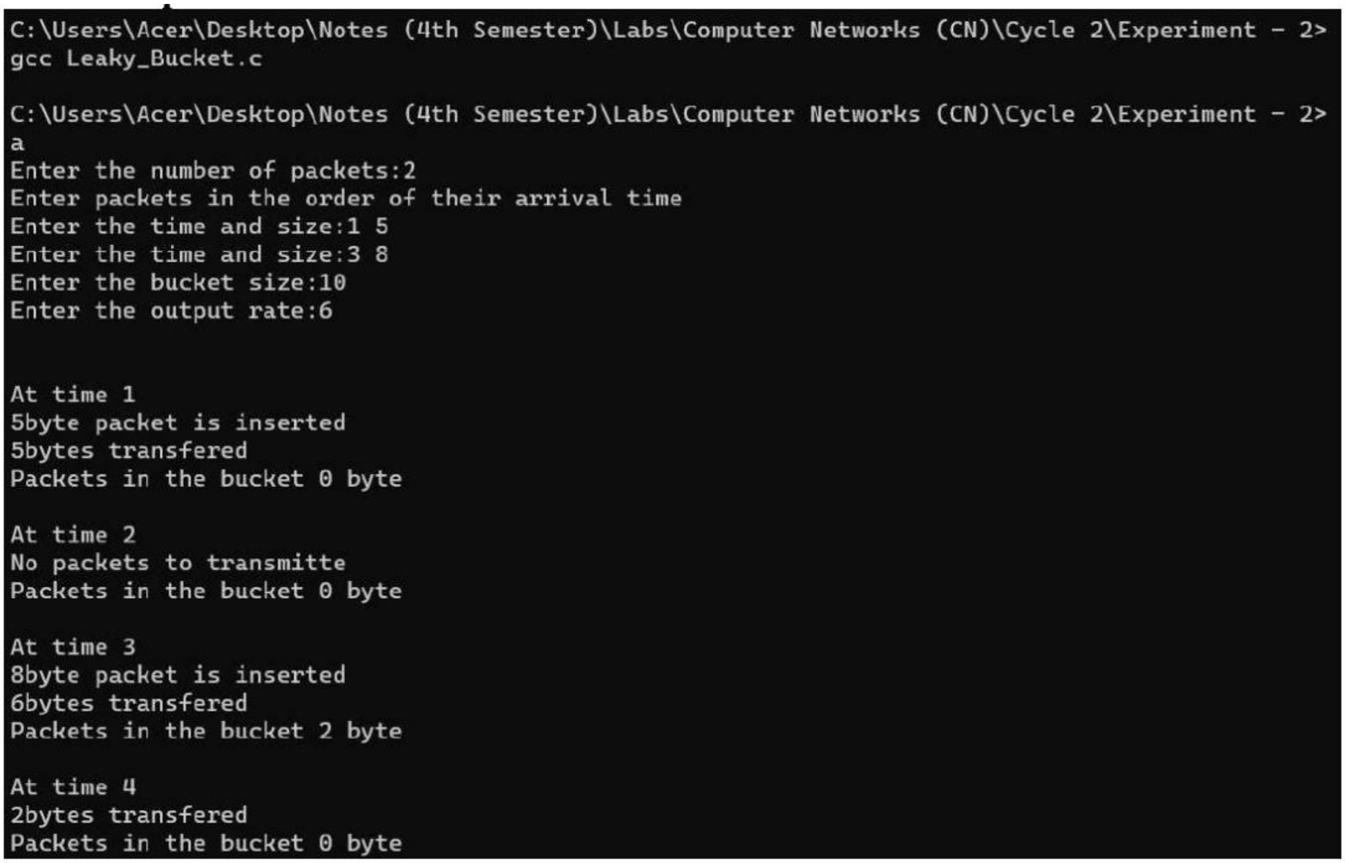
```
C:\Users\Acer\Desktop\Notes (4th Semester)\Labs\Computer Networks (CN)\Cycle 2\Experiment - 1>a
Enter Bit string: 110110101
Enter generating polynomial (16 bits): 1101100000000001
Generating Polynomial: 1101100000000001
Modified Data is: 110110101000000000000000
Checksum is: 101000100000110
Final Codeword is: 110110101101000100000110
Test Error detection
1(Yes) / 0(No)? : 0
No Error Detected.

C:\Users\Acer\Desktop\Notes (4th Semester)\Labs\Computer Networks (CN)\Cycle 2\Experiment - 1>a
Enter Bit string: 110110101
Enter generating polynomial (16 bits): 1101100000000001
Generating Polynomial: 1101100000000001
Modified Data is: 110110101000000000000000
Checksum is: 101000100000110
Final Codeword is: 110110101101000100000110
Test Error detection
1(Yes) / 0(No)? : 1
Enter position to insert an error: 7
Erroneous data: 110110111101000100000110
Error detected.
```

Lab 14

Write a program for congestion control using Leaky bucket algorithm.

Code:

```
#include <stdio.h>
#include <stdlib.h>
struct packet
{
    int time;
    int size;
} p[50];

int main()
{
    int i, n, m, k = 0;
    int bsize, bfilled, outrate;
    printf("Enter the number of packets:");
    scanf("%d", &n);
    printf("Enter packets in the order of their arrival time\n");
    for (i = 0; i < n; i++)
    {
        printf("Enter the time and size:");
        scanf("%d%d", &p[i].time, &p[i].size);
    }
    printf("Enter the bucket size:");
    scanf("%d", &bsize);
    printf("Enter the output rate:");
    scanf("%d", &outrate);

    m = p[n - 1].time;
    i = 1;
    k = 0;
    bfilled = 0;
    while (i <= m || bfilled != 0)
    {
        printf("\n\nAt time %d", i);

        if (p[k].time == i)
        {
            if (bsize >= bfilled + p[k].size)
            {
                bfilled = bfilled + p[k].size;
                printf("\n%dbyte packet is inserted", p[k].size);
        k = k + 1;
      }
      else
      {
        printf("\n%dbyte packet is discarded", p[k].size);
        k = k + 1;
      }

    }

  if (bfilled == 0)
  {
    printf("\nNo packets to transmitte");
  }
  else if (bfilled >= outrate)
  {
    bfilled = bfilled - outrate;
    printf("\n%dbytes transfered", outrate);
  }

  else
  {
    printf("\n%dbytes transfered", bfilled);
    bfilled = 0;
  }
  printf("\nPackets in the bucket %d byte", bfilled);
  i++;
  }
  return 0;
}
```



```
C:\Users\Acer\Desktop\Notes (4th Semester)\Labs\Computer Networks (CN)\Cycle 2\Experiment - 2>
gcc Leaky_Bucket.c

C:\Users\Acer\Desktop\Notes (4th Semester)\Labs\Computer Networks (CN)\Cycle 2\Experiment - 2>
a
Enter the number of packets:2
Enter packets in the order of their arrival time
Enter the time and size:1 5
Enter the time and size:3 8
Enter the bucket size:10
Enter the output rate:6


At time 1
5byte packet is inserted
5bytes transfered
Packets in the bucket 0 byte

At time 2
No packets to transmitte
Packets in the bucket 0 byte

At time 3
8byte packet is inserted
6bytes transfered
Packets in the bucket 2 byte

At time 4
2bytes transfered
Packets in the bucket 0 byte
```

Lab 15

Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

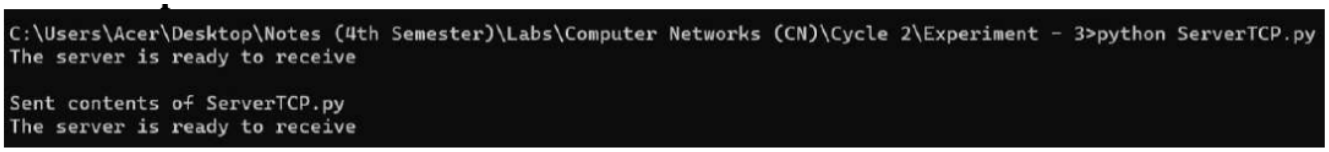
Code:

ClientTCP.py

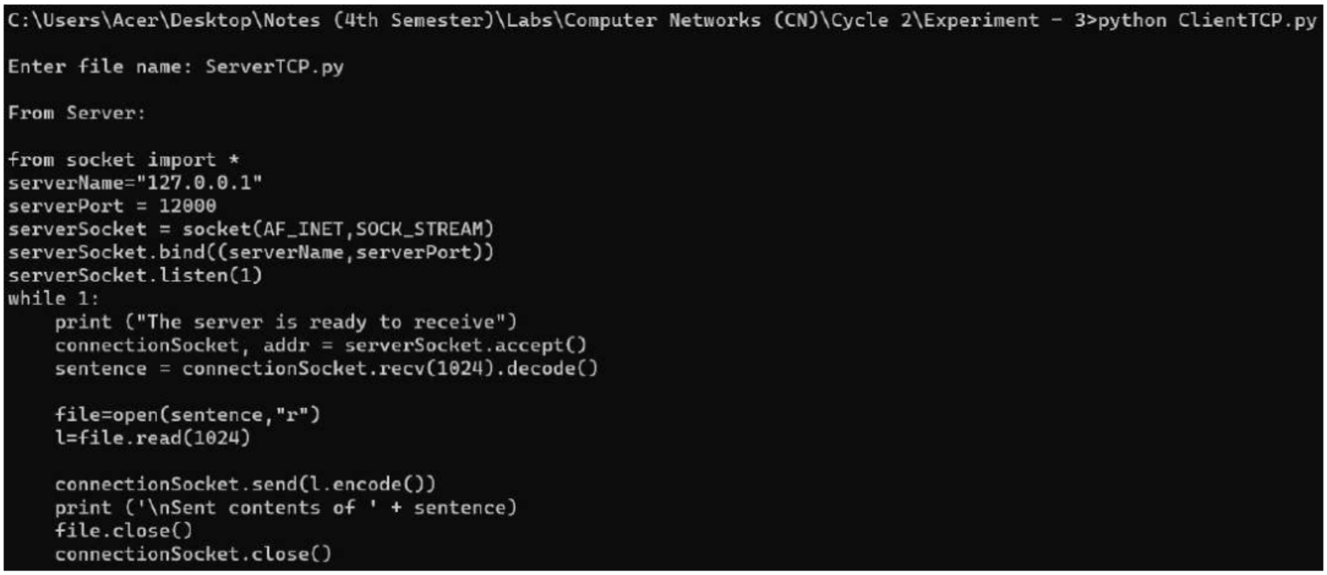
```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

ServerTCP.py

```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    file=open(sentence,"r")
    l=file.read(1024)

    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
   connectionSocket.close()
```



```
C:\Users\Acer\Desktop\Notes (4th Semester)\Labs\Computer Networks (CN)\Cycle 2\Experiment - 3>python ServerTCP.py
The server is ready to receive

Sent contents of ServerTCP.py
The server is ready to receive
```



```
C:\Users\Acer\Desktop\Notes (4th Semester)\Labs\Computer Networks (CN)\Cycle 2\Experiment - 3>python ClientTCP.py

Enter file name: ServerTCP.py

From Server:

from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    file=open(sentence,"r")
    l=file.read(1024)

    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

Lab 16

Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

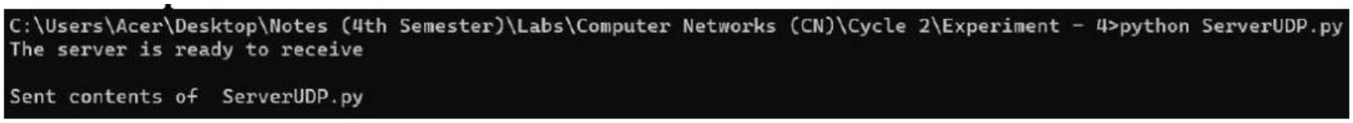
Code:

ClientUDP.py

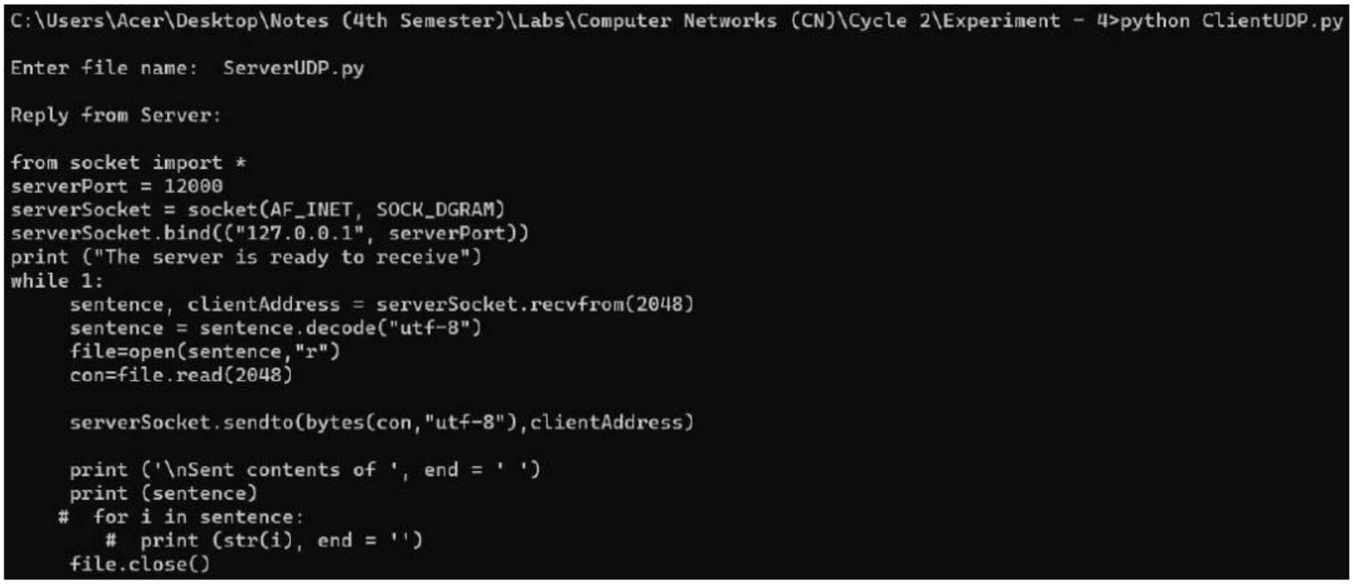
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

sentence = input("\nEnter file name:  ")

clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))

filecontents,serverAddress = clientSocket.recvfrom(2048)
print ('\nReply from Server:\n')
print (filecontents.decode("utf-8"))
# for i in filecontents:
   # print(str(i), end = '')
clientSocket.close()
clientSocket.close()
```

ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)

    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)

    print ('\nSent contents of ', end = ' ')
    print (sentence)
   #  for i in sentence:
      #  print (str(i), end = '')
    file.close()
```



```
C:\Users\Acer\Desktop\Notes (4th Semester)\Labs\Computer Networks (CN)\Cycle 2\Experiment - 4>python ServerUDP.py
The server is ready to receive

Sent contents of  ServerUDP.py
```



```
C:\Users\Acer\Desktop\Notes (4th Semester)\Labs\Computer Networks (CN)\Cycle 2\Experiment - 4>python ClientUDP.py

Enter file name:  ServerUDP.py

Reply from Server:

from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
     sentence, clientAddress = serverSocket.recvfrom(2048)
     sentence = sentence.decode("utf-8")
     file=open(sentence,"r")
     con=file.read(2048)

     serverSocket.sendto(bytes(con,"utf-8"),clientAddress)

     print ('\nSent contents of ', end = ' ')
     print (sentence)
    #  for i in sentence:
        #  print (str(i), end = '')
     file.close()
```