CYCLE 2-1BM21CS203

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

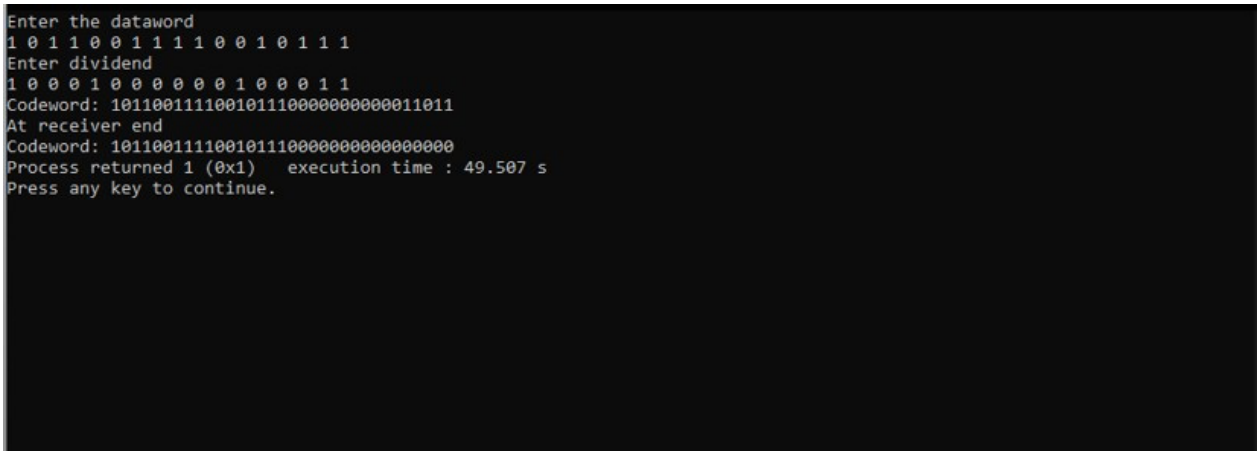
Code:

```
#include<stdio.h>
int arr[17];

void xor(int x[], int y[])
{
    int k=0;
    for(int i=1;i<16;i++)
    {
        if(x[i]==y[i])
            arr[k++]=0;
        else
            arr[i]=1;
    }
}

void main()
{
    int dd[17],div[33],ze[17],i,k;

    printf("Enter the dataword \n");
```

```
for(i=0;i<17;i++)
    scanf("%d",&div[i]);
for(i=i;i<33;i++)
    div[i]=0;

for(i=0;i<17;i++)ze[i]=0;
printf("Enter dividend \n");
for(i=0;i<17;i++)
    scanf("%d",&dd[i]);

i=0;
k=0;
    for(i=i;i<17;i++)
        arr[k++]=div[i];
while(i<33)
{
    if(arr[0]==0)
        xor(arr,ze);
    else
        xor(arr,dd);

    arr[16]=div[i++];

}
k=0;
for(i=17;i<33;i++)div[i]=arr[k++];
```

```
    printf("Codeword:  ");
        for(i=0;i<33;i++)
          printf("%d",div[i]);

    for(i=0;i<17;i++)
        arr[i]=0;

     printf("\nAt receiver end \n");

      k=0;
        for(i=i;i<17;i++)
            arr[k++]=div[i];
    while(i<33)
    {
        if(arr[0]==0)
            xor(arr,ze);
        else
            xor(arr,dd);

        arr[16]=div[i++];
}
k=0;
    for(i=17;i<33;i++)div[i]=arr[k++];
    printf("Codeword:  ");
        for(i=0;i<33;i++)
            printf("%d",div[i]);
 }
```

Output:


Enter the dataword
1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1
Enter dividend
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1
Codeword: 10110011110010111000000000011011
At receiver end
Codeword: 10110011110010111000000000000000
Process returned 1 (0x1)   execution time : 49.507 s
Press any key to continue.

Question: Write a program for congestion control using Leaky bucket

algorithm.

Code:

```
#include <stdio.h>
#include <stdlib.h> // Include this for the rand() function


int main()
{
   int buckets, outlets, k = 1, num, remaining;


   printf("Enter Bucket size and outstream size\n");
   scanf("%d %d", &buckets, &outlets);
   remaining = buckets;


   while (k)
   {
      num = rand() % 1000; // Generate a random number between 0
      and 999if(num < remaining)
      {
         remaining = remaining - num;
         printf("Packet of %d bytes accepted\n", num); // Added missing variable
      }
      else
      {
         printf("Packet of %d bytes is discarded\n", num);
      }
      if (buckets - remaining > outlets)
      {
         remaining += outlets; // Fixed the calculation
      }
      else
         remaining = buckets;
      printf("Remaining bytes: %d \n", remaining);
      printf("If you want to stop input, press 0, otherwise, press 1\n");
      scanf("%d", &k);
```

```
    }

    while (remaining < buckets) // Fixed the condition
{       if (buckets - remaining > outlets)
        {
            remaining += outlets; // Fixed the calculation
        }
        else
            remaining = buckets;
        printf("Remaining bytes: %d \n", remaining);
    }
    return 0; // Added a return statement to indicate successful completion
 }
```

Output:

```
PS D:\VS Code> cd "d:\VS Code\OS\" ; if ($?) { gcc bucket.c -o bucket } ; if ($?) { .\bucket }
Enter Bucket size and outstream size
2000
100
Packet of 41 bytes accepted
Remaining bytes: 2000
If you want to stop input, press 0, otherwise, press 1
1
Packet of 467 bytes accepted
Remaining bytes: 1633
If you want to stop input, press 0, otherwise, press 1
1
Packet of 334 bytes accepted
Remaining bytes: 1399
If you want to stop input, press 0, otherwise, press 1
1
Packet of 500 bytes accepted
Remaining bytes: 999
If you want to stop input, press 0, otherwise, press 1
1
Packet of 169 bytes accepted
Remaining bytes: 930
If you want to stop input, press 0, otherwise, press 1
1
Packet of 724 bytes accepted
Remaining bytes: 306
If you want to stop input, press 0, otherwise, press 1
1
Packet of 478 bytes is discarded
Remaining bytes: 406
If you want to stop input, press 0, otherwise, press 1
1
Packet of 358 bytes accepted
Remaining bytes: 148
If you want to stop input, press 0, otherwise, press 1
1
Packet of 962 bytes is discarded
Remaining bytes: 248
If you want to stop input, press 0, otherwise, press 1
0
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
```


Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
Remaining bytes: 848
Remaining bytes: 948
Remaining bytes: 1048
Remaining bytes: 1148
Remaining bytes: 1248
Remaining bytes: 1348
Remaining bytes: 1448
Remaining bytes: 1548
Remaining bytes: 1648
Remaining bytes: 1748
Remaining bytes: 1848
Remaining bytes: 1948
Remaining bytes: 2000
PS D:\VS Code\OS>

QUESTION:

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 requestedfile if present.

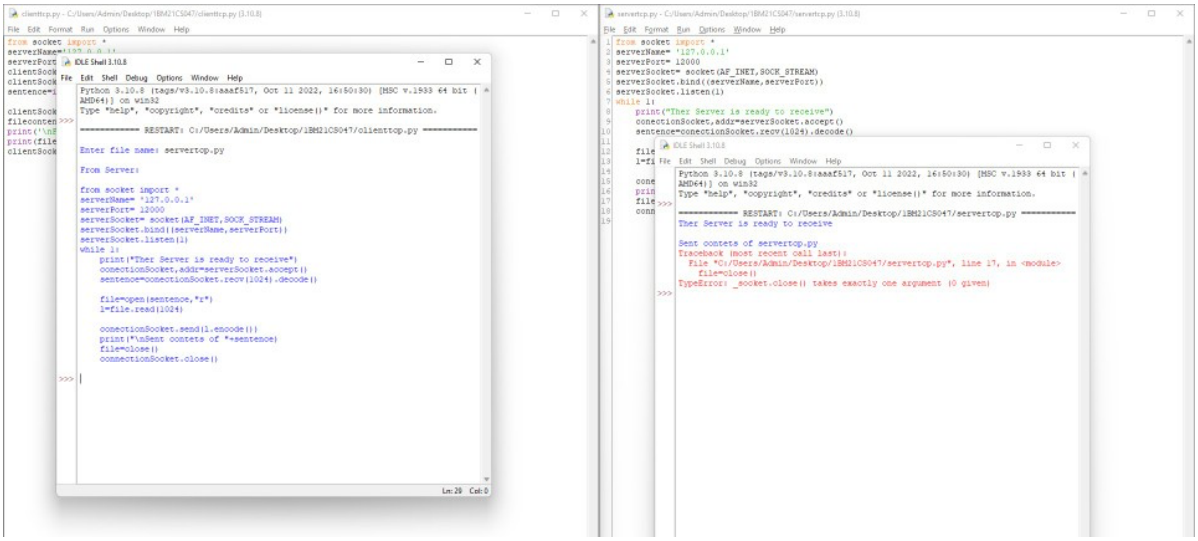
Code:

Server.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("Ther Server is ready to receive") conectionSocket,addr=serverSocket.accept()
    sentence=conectionSocket.recv(1024).decode()

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

    conectionSocket.send(l.encode()
    ) print("\nSent contets of
    "+sentence)file=close()
    connectionSocket.close()
```

Client.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()
```


clienttcp.py - C:/Users/Admin/Desktop/1BM21CS047/clienttcp.py (3.10.8)
File Edit Format Run Options Window Help
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
========== RESTART: C:/Users/Admin/Desktop/1BM21CS047/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('Ther 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 contets of '+sentence)
file=close()
connectionSocket.close()
Ln: 29 Col: 0
servertcp.py - C:/Users/Admin/Desktop/1BM21CS047/servertcp.py (3.10.8)
File Edit Format Run Options Window Help
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("Ther Server is ready to receive")
connectionSocket,addr=serverSocket.accept()
sentence=connectionSocket.recv(1024).decode()
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
========== RESTART: C:/Users/Admin/Desktop/1BM21CS047/servertcp.py ==========
Ther Server is ready to receive
Sent contets of servertcp.py
Traceback (most recent call last):
File "C:/Users/Admin/Desktop/1BM21CS047/servertcp.py", line 17, in <module>
file=close()
TypeError: _socket.close() takes exactly one argument (0 given)

Question: 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 requestedfile if present.

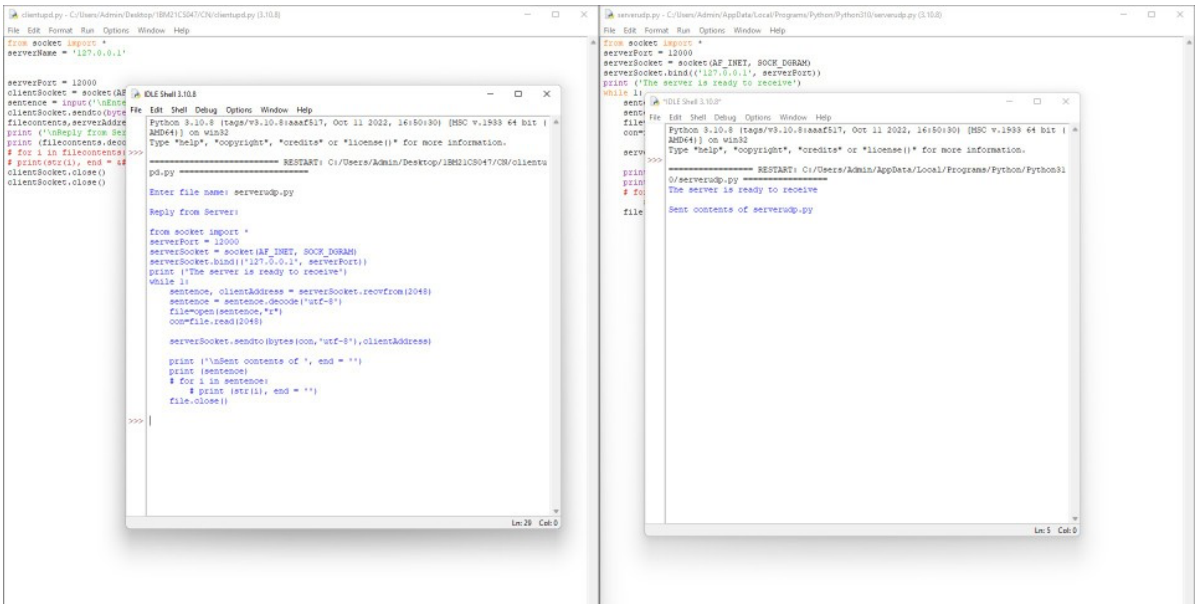
Code:

Server.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()
```

Client.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'))clientSocket.close()
clientSocket.close()
```


clientupd.py - C:/Users/Admin/Desktop/18M21CS047/CN/clientupd.py (3.10.8)
File Edit Format Run Options Window Help
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF
sentence = input('\nEnte
clientSocket.sendto(byte
filecontents,serverAddre
print ('\nReply from Ser
print (filecontents.deco
# for i in filecontents:
# print(str(i), end = ''
clientSocket.close()
clientSocket.close()
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
========== RESTART: C:/Users/Admin/Desktop/18M21CS047/CN/clientu
pd.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()
>>>
Ln: 29 Col: 0
serverudp.py - C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverudp.py (3.10.8)
File Edit Format Run Options Window Help
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()
IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
========== RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python31
0/serverudp.py ==========
The server is ready to receive
Sent contents of serverudp.py
Ln: 5 Col: 0