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;

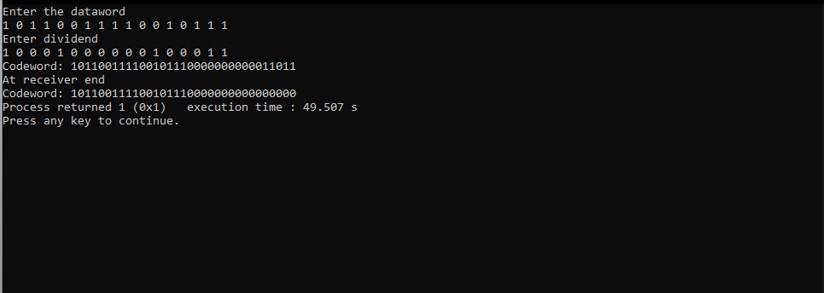
for(i=17;i<33;i++)div[i]=arr[k++];

printf("Codeword: "); for(i=0;i<33;i++)

printf("%d",div[i]);

}

Output:



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;

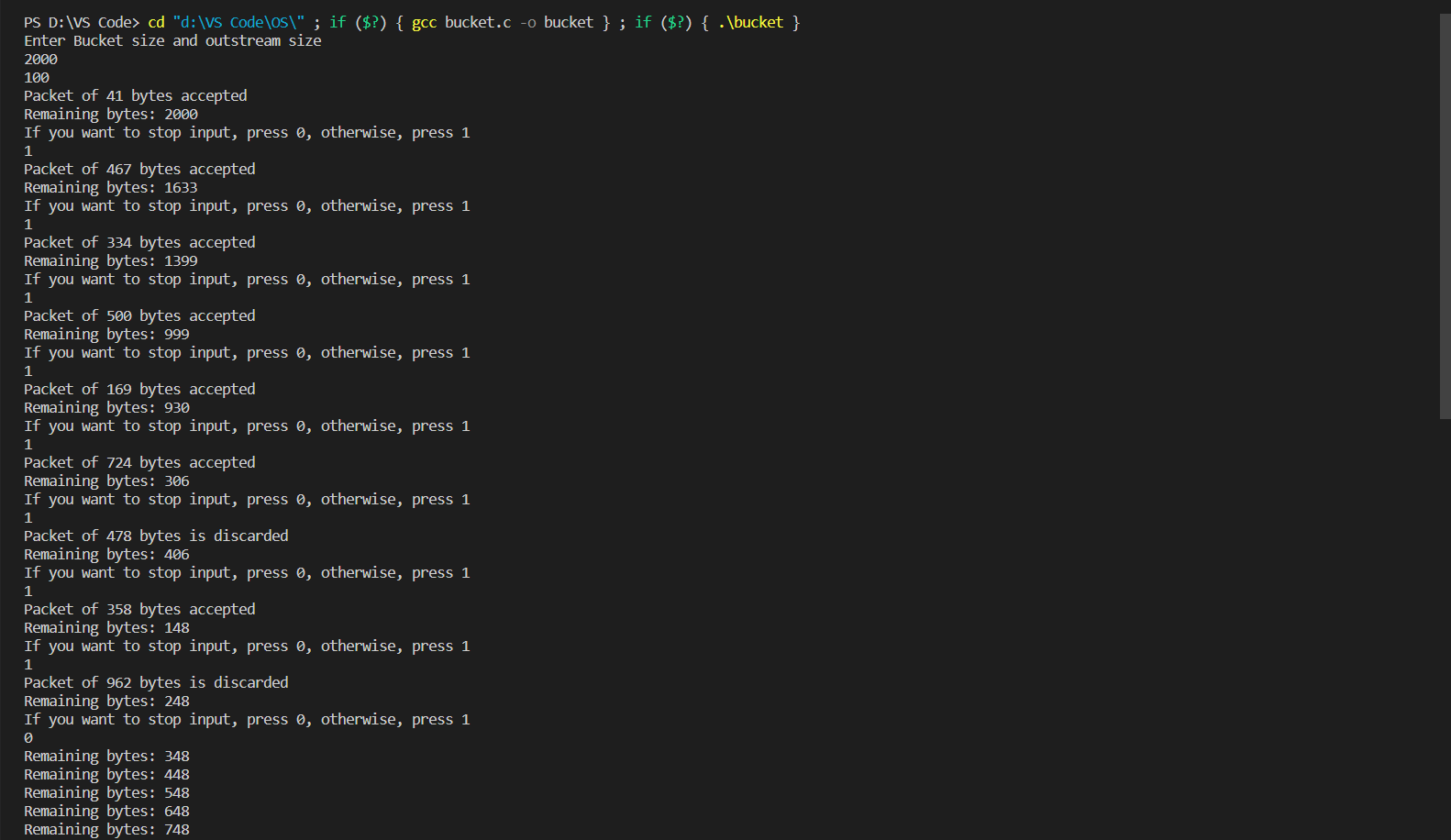
printf("Remaining bytes: %d \n", remaining);

}

return 0; // Added a return statement to indicate successful completion

}

Output:





**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 requested file 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

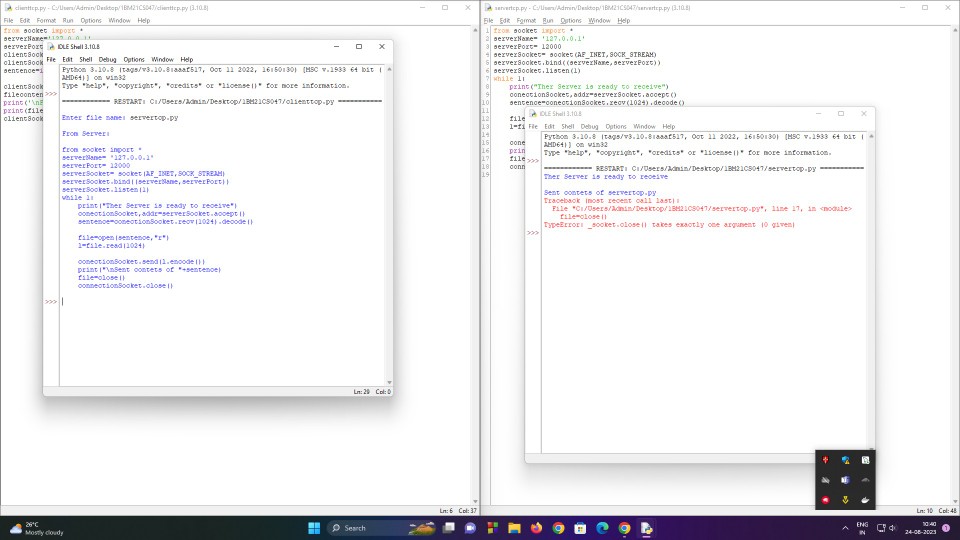
from socket import \* serverName='127.0.0. 1'serverPort=12000

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()



**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.**

**Question:**

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

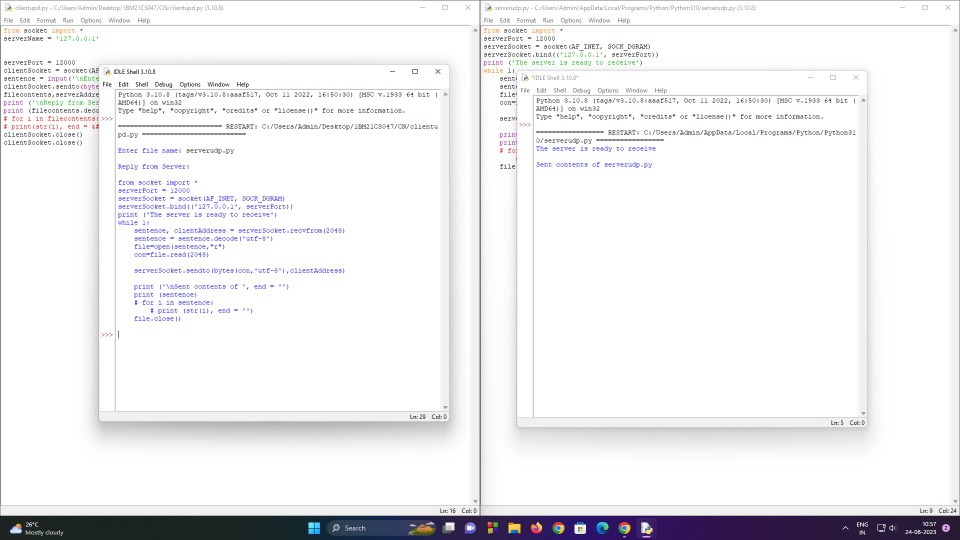
from socket import \* serverName = '127.0.0.1'

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()

****