**2. Implement on Data set of characters the three CRC polynomials – CRC 12, CRC-16 and CRC CCIP.**

**crc\_rec.py:**

import socket

def xor(a, b):

result = []

for i in range(1, len(b)):

if a[i] == b[i]:

result.append('0')

else:

result.append('1')

return ''.join(result)

def mod2div(divident, divisor):

pick = len(divisor)

tmp = divident[0: pick]

while pick < len(divident):

if tmp[0] == '1':

tmp = xor(divisor, tmp) + divident[pick]

else:

tmp = xor('0'\*pick, tmp) + divident[pick]

pick += 1

if tmp[0] == '1':

tmp = xor(divisor, tmp)

else:

tmp = xor('0'\*pick, tmp)

checkword = tmp

return checkword

def decodeData(data, key):

l\_key = len(key)

appended\_data = data

remainder = mod2div(appended\_data, key)

return remainder

s = socket.socket()

print("Socket successfully created")

port = 1240

s.bind(('', port))

print("socket binded to %s" % (port))

s.listen(5)

print("socket is listening")

while True:

c, addr = s.accept()

print('Got connection from', addr)

data = c.recv(1024)

data = data.decode()

data,key = data.split('')

print("Received encoded data in binary format :", data)

print("Received key :",key)

if not data:

break

ans = decodeData(data, key)

print("Remainder after decoding is->"+ans)

temp = "0" \* (len(key) - 1)

if ans == temp:

msg = data[:-(len(key)-1)]

l = len(msg)

x = 0

y = 7

z = ""

while x < l:

m = msg[x:y]

z = z + chr(int(m,2))

x += 7

y += 7

print(z)

c.sendto(("THANK you Data ->"+data +

" Received No error FOUND").encode(), ('127.0.0.1', port))

else:

c.sendto(("Error in data").encode(), ('127.0.0.1', port))

c.close()

**crc\_sender.py:**

import socket

def xor(a, b):

result = []

for i in range(1, len(b)):

if a[i] == b[i]:

result.append('0')

else:

result.append('1')

return ''.join(result)

def mod2div(divident, divisor):

pick = len(divisor)

tmp = divident[0 : pick]

while pick < len(divident):

if tmp[0] == '1':

tmp = xor(divisor, tmp) + divident[pick]

else:

tmp = xor('0'\*pick, tmp) + divident[pick]

pick += 1

if tmp[0] == '1':

tmp = xor(divisor, tmp)

else:

tmp = xor('0'\*pick, tmp)

checkword = tmp

return checkword

def encodeData(data, key):

l\_key = len(key)

appended\_data = data + '0'\*(l\_key-1)

remainder = mod2div(appended\_data, key)

codeword = data + remainder

return codeword

def crcfind():

ans = encodeData(data,key)

print("Encoded data to be sent to server in binary format :",ans)

ans = ans +""+key

s.sendto(ans.encode(),('127.0.0.1', port))

print("Received feedback from server :",s.recv(1024).decode())

s = socket.socket()

port = 1240

s.connect(('127.0.0.1', port))

input\_string = input("Enter data you want to send->")

data =(''.join(format(ord(x), 'b') for x in input\_string))

print("Entered data in binary format :",data)

while True:

print("Chose which CRC Technique u want to use: ")

print("1.CRC-12")

print("2.CRC-16")

print("3.exit")

n = int(input())

if n == 1:

key = input("enter key for CRC-12 (The key should start and end with 1 and contains 13 digits) : ")

crcfind()

elif n == 2:

key = input("enter key for CRC-16 (The key should start and end with 1 and contains 17 digits) : ")

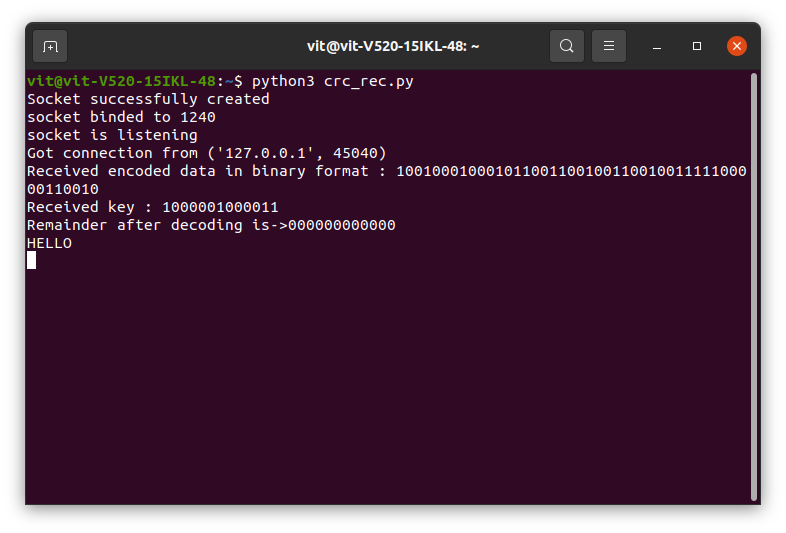
crcfind()

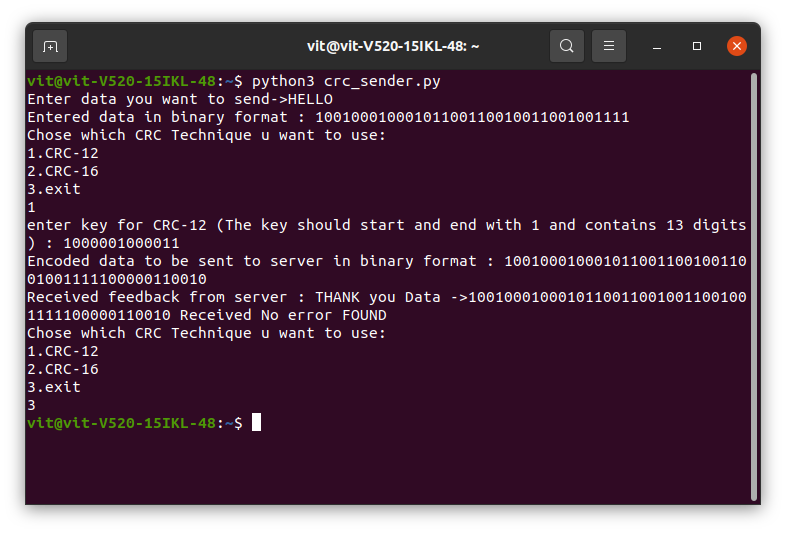
else:

break

s.close()

**Output:** First run the server program in one terminal and then run client program in another terminal.

****

****