CRC - 16 bit

1BM18CS0

Sender side

```
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 temp [0] == '1':
        temp = xor (divisor, tmp) + divident[pick].
    else:
        temp = xor ('0' * pick, temp) + 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' *(key - 1).
    remainder = mod2div (appended_data, key)
    codeword = data + remainder.
    return codeword.
  s = socket.socket().
  port = 12345.
  s.connect(('127.0.0.1', port)).
  input_string = raw_input ("Enter data you want to.
   send-> ")
   data = (''.join (format (ord(x), 'b') for x in input_string
   print data.
   key = "1001"
   ans = encodeData (data, key).
   print (ans).
    s.sendall (ans).
    print : s.recv (1024).
    s.close()
```

Reciever side

```
 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 decode_Data (data, key):
      L_key = len (key).
      append_data = data + '0' * (L_key - 1)
      remainder = mod2div (append_data, key).
    return remainder.

s = socket. socket()
print ("socket successfully created").
port = 12345
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)
    print (data)
    if not data:
        break.
   key = "100"
  ans = decodeData (data, key)
  print ("Remainder ~~is all zeros then no error occured)~~
                    after decoding is →" +ans)
   temp = "0" * ((len(key) -1)
   if ans == temp:
     c.sendall ("Thank you data → "+data + " Received
                    No error found").
   else
     c.sendall ("Error in data").
  c.close ()
```