**PS 10 CRC**

**By**

**Abishek.A 20pw01**

class CRC:

def \_\_init\_\_(self):

self.cdw = ''

def xor(self,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 crc(self,message, key):

pick = len(key)

tmp = message[:pick]

while pick < len(message):

if tmp[0] == '1':

tmp = self.xor(key,tmp)+message[pick]

else:

tmp = self.xor('0'\*pick,tmp) + message[pick]

pick+=1

if tmp[0] == "1":

tmp = self.xor(key,tmp)

else:

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

checkword = tmp

return checkword

def encodedData(self,data,key):

l\_key = len(key)

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

remainder = self.crc(append\_data,key)

codeword = data+remainder

self.cdw += codeword

print("Remainder: " ,remainder)

print("Data: " ,codeword)

def reciverSide(self,key,data):

r = self.crc(data,key)

size = len(key)

print(r)

if r == size\*0:

print("No Error")

else:

print("Error")

data=input("Data: ")

#data = '1001'

key = input("Key: ")

#key = '1101'

c = CRC()

c.encodedData(data,key)

c.reciverSide(c.cdw,key)

print(c.cdw)