**[2CEIT503: COMPUTER NETWORK]**

Practical: 5



**AIM: Write a program to implement various Error Detection Mechanisms.**

**a. find minimum hamming distance**

**b. Checksum**

**c. CRC**



**Department of Computer Engineering/Information Technology**

**Q.1 find minimum hamming distance**

no\_of\_codeword=int(input("Enter no of code word: "))

codeword\_bit=int(input("Enter no of bit in one data: "))

codeword=[]

counter=[]

for i in range(0,no\_of\_codeword):

    print("Enter ",i," Codeword bit: ")

    d=list(input("Enter bit: "))

    codeword.append(d)

print(codeword)

for i in range(0,len(codeword)-1):

    for j in range(i+1,len(codeword)):

        c=-1

        for k in range(0, codeword\_bit):

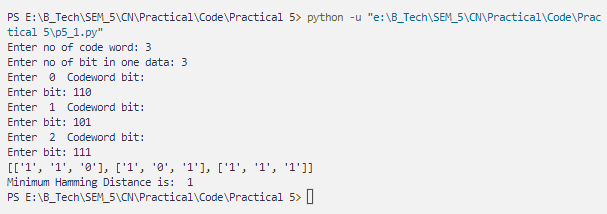
            if(int(codeword[i][k]) != int(codeword[j][k])):

                c=c+1

        counter.append(c+1)

print("Minimum Hamming Distance is: ",min(counter))

**Output:**

****

**Q.2 Checksum**

k=int(input("No of Part: "))

m=int(input("Enter no of bit in one part: "))

dataword=[]

sum="0"

#Sender Side Process

print("====Enter Data At Sender Side====")

for i in range(k):

    d=input("Data "+str(i+1)+": ")

    sum=bin(int(sum,2)+int(d, 2)).replace("0b", "")

    dataword.append(d)

print("Sender side Data: ",dataword)

print("Sum is: ",sum)

while(True):

    if(len(sum)>m):

        l=len(sum)-m

        sum = bin(int(sum[:l],2)+int(sum[l:],2)).replace("0b","")

    else:

        break

print("Wrap Sum is: ",sum)

checksum=""

for i in sum:

    if(i=='1'):

        checksum=checksum+"0"

    else:

        checksum=checksum+"1"

print("Checksum is: ",checksum)

# Receiver Side Process

print("====Enter Data At Receiver Side====")

for i in range(k):

    d=input("Data "+str(i+1)+": ")

    dataword[i]=d

print("Receiver Side Data is: ",dataword)

dataword.append(checksum)

sum="0"

for i in range(len(dataword)):

    sum=bin(int(sum,2)+int(dataword[i],2)).replace("0b","")

if(len(sum)>m):

    l=len(sum)-m

    sum = bin(int(sum[:l],2)+int(sum[l:],2)).replace("0b","")

f=1

result=""

for i in sum:

    if(i=='1'):

        result=result+"0"

    else:

        f=0

        result=result+"1"

if(f==1):

    print("Data is Correct")

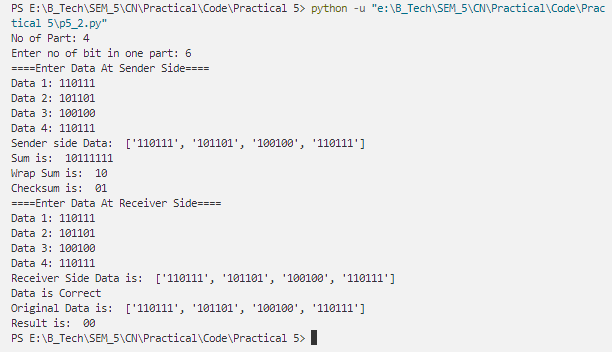
    print("Original Data is: ",dataword[0:len(dataword)-1])

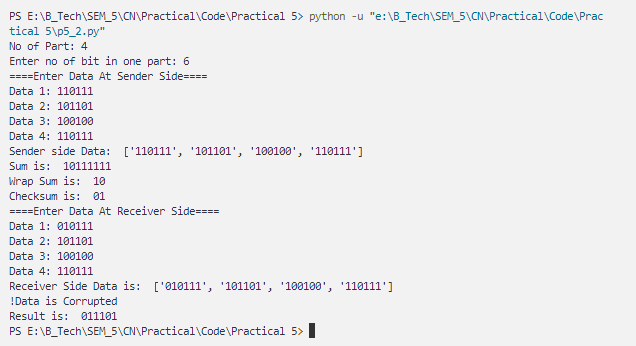
else:

    print("!Data is Corrupted")

print("Result is: ",result)

**Output:**

****

****

**Q.3 CRC**

def xor(a, b):

    result = []

    for i in range(len(b)):

        if a[i] == b[i]:

            result.append('0')

        else:

            result.append('1')

    return ''.join(result)

def divmodulo(data,dim):

    index=0

    while(True):

        if(data[index]=="0"):

            index=index+1

        else:

            break

    r = xor(data[index:index+len(dim)],dim)

    c=0

    index=index+len(dim)

    for i in range(len(r)):

        if(r[i]=='0'):

            r=r+data[index]

            index=index+1

            c=c+1

        else:

            break

    r=r[c:]

    while(index<len(data)):

        r = xor(r,dim)

        c=0

        for i in range(len(r)):

            if(r[i]=='0'):

                if(index<len(data)):

                    r=r+data[index]

                    index=index+1

                    c=c+1

                else:

                    return r

            else:

                break

        r=r[c:]

    r=xor(r,dim)

    return r

#Process at Sender Side

data=input("Enter Data at Sender Side: ")

dim = input("Enter Dim: ")

print("Data is: ",data)

print("Divisor is: ",dim)

crc=""

for i in range(len(dim)-1):

    crc=crc+"0"

dataword=data+crc

sender\_reminder = divmodulo(dataword,dim)

print("Dataword+CRC is: ",dataword)

sender\_reminder=sender\_reminder[len(sender\_reminder)%len(crc):]

print("Reminder at Sender side: ",sender\_reminder)

codeword = bin(int(dataword,2)+int(sender\_reminder,2)).replace("0b","")

print("Codeword is: ",codeword)

#Process At receiver Side

r\_data=input("Enter Data at Receiver side: ")

receiver\_reminder=divmodulo(r\_data,dim)

receiver\_reminder=receiver\_reminder[len(receiver\_reminder)%len(crc):]

print("Reminder at Receiver side: ",receiver\_reminder)

if("1" in receiver\_reminder):

    print("Something Error with Message")

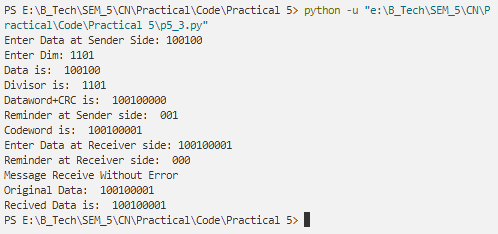
else:

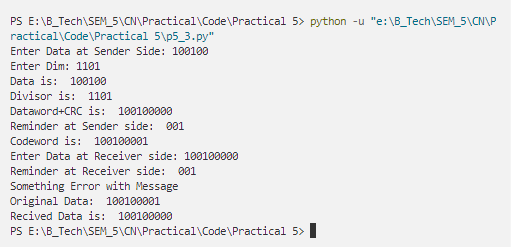
    print("Message Receive Without Error")

print("Original Data: ",codeword)

print("Recived Data is: ",r\_data)

**Output:**

****

****