[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:

```
PS E:\B_Tech\SEM_5\CN\Practical\Code\Practical 5> python -u "e:\B_Tech\SEM_5\CN\Practical\Code\Prac
tical 5\p5_1.py"
Enter no of code word: 3
Enter no of bit in one data: 3
Enter  0  Codeword bit:
Enter bit: 110
Enter  1  Codeword bit:
Enter bit: 101
Enter  2  Codeword bit:
Enter bit: 111
[['1', '1', '0'], ['1', '0', '1'], ['1', '1', '1']]
Minimum Hamming Distance is:  1
PS E:\B_Tech\SEM_5\CN\Practical\Code\Practical 5>
```

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:

```
PS E:\B_Tech\SEM_5\CN\Practical\Code\Practical 5> python -u "e:\B_Tech\SEM_5\CN\Practical\Code\Prac
tical 5\p5_2.py"
No of Part: 4
Enter no of bit in one part: 6
====Enter Data At Sender Side====
Data 1: 110111
Data 2: 101101
Data 3: 100100
Data 4: 110111
Sender side Data:  ['110111', '101101', '100100', '110111']
Sum is:  10111111
Wrap Sum is:  10
Checksum is:  01
====Enter Data At Receiver Side====
Data 1: 110111
Data 2: 101101
Data 3: 100100
Data 4: 110111
Receiver Side Data is:  ['110111', '101101', '100100', '110111']
Data is Correct
Original Data is:  ['110111', '101101', '100100', '110111']
Result is:  00
PS E:\B_Tech\SEM_5\CN\Practical\Code\Practical 5>
```

```
PS E:\B_Tech\SEM_5\CN\Practical\Code\Practical 5> python -u "e:\B_Tech\SEM_5\CN\Practical\Code\Prac
tical 5\p5_2.py"
No of Part: 4
Enter no of bit in one part: 6
====Enter Data At Sender Side====
Data 1: 110111
Data 2: 101101
Data 3: 100100
Data 4: 110111
Sender side Data:  ['110111', '101101', '100100', '110111']
Sum is:  10111111
Wrap Sum is:  10
Checksum is:  01
====Enter Data At Receiver Side====
Data 1: 010111
Data 2: 101101
Data 3: 100100
Data 4: 110111
Receiver Side Data is:  ['010111', '101101', '100100', '110111']
!Data is Corrupted
Result is:  011101
PS E:\B_Tech\SEM_5\CN\Practical\Code\Practical 5>
```

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:

```
PS E:\B_Tech\SEM_5\CN\Practical\Code\Practical 5> python -u "e:\B_Tech\SEM_5\CN\P
ractical\Code\Practical 5\p5_3.py"
Enter Data at Sender Side: 100100
Enter Dim: 1101
Data is:  100100
Divisor is:  1101
Dataword+CRC is:  100100000
Reminder at Sender side:  001
Codeword is:  100100001
Enter Data at Receiver side: 100100001
Reminder at Receiver side:  000
Message Receive Without Error
Original Data:  100100001
Recived Data is:  100100001
PS E:\B_Tech\SEM_5\CN\Practical\Code\Practical 5>
```

```
PS E:\B_Tech\SEM_5\CN\Practical\Code\Practical 5> python -u "e:\B_Tech\SEM_5\CN\P
ractical\Code\Practical 5\p5_3.py"
Enter Data at Sender Side: 100100
Enter Dim: 1101
Data is:  100100
Divisor is:  1101
Dataword+CRC is:  100100000
Reminder at Sender side:  001
Codeword is:  100100001
Enter Data at Receiver side: 100100000
Reminder at Receiver side:  001
Something Error with Message
Original Data:  100100001
Recived Data is:  100100000
PS E:\B_Tech\SEM_5\CN\Practical\Code\Practical 5>
```