Name: Vinit Shah	SAP ID:60004220097
Date of Performance:21-08-24	Date of Submission:28-08-24

Experiment No: 3

Aim: Study and Implement Error Detection and Correction using CRC and Hamming Code

Theory:

1)CRC

Program:

```
divisor=input("Enter the key:")
d=[]
divisor_len=len(divisor)
for i in divisor:
    d.append(i)


dividend=input("Enter the data word to be sent:")
dividend1=dividend
di1=[]
for i in dividend:
    di1.append(i)
dividend+='000'
di=[]
dividend_len=len(dividend)
for i in dividend:
    di.append(i)


count=0
string=[]
quotient=[]
```

```
for i in range(divisor_len):
    string.append(di[count])
    count+=1


def xor(a,b):
    if a==b:
        return '0'
    else:
        return '1'


final=[]


def first(a):
    if a[0]=='1':
        for i in range(len(a)):
            result=xor(a[i],d[i])
            final.append(result)
        quotient.append('1')
    else:
        for i in range(len(a)):
            final.append(string[i])
        quotient.append('0')


first(string)


for i in range(count,dividend_len):
    string=[]
    for j in range(1,len(final)):
        string.append(final[j])
```

```
    string.append(di[i])


    final=[]
    first(string)


remainder=[]
for i in range(len(di1)):
    remainder.append(di1[i])
for i in range(1,len(final)):
    remainder.append(final[i])


result=''.join(remainder)
print("Code sent:"+result)
```

Screenshots:

```
>>>
======================== RESTART: C:/Users/vidhi shah/Desktop/python/Sem 5 cn/Exp03-CRC.py ========================
Enter the key:1101
Enter the data word to be sent:100100
Code sent:100100001
>>>
```

2)Hamming Code:

Program:

```
code=input("Enter the code to be transmitted:")
c=[]
for i in code:
    c.append(i)
no_of_bits=len(code)
#print(no_of_bits)


def paritybits():
    i=0
```

```
    while i<no_of_bits:
        a=pow(2,i)
        #print(a)
        b=no_of_bits+i+1
        #print(b)
        if a>=b:
            break
        else:
            i+=1
    return i


no_of_paritybits=paritybits()
#print(no_of_paritybits)


h=[]
count=0
index=no_of_bits-1
for i in range(no_of_bits+no_of_paritybits):
    a=pow(2,count)
    if (i+1)==a:
        h.append('')
        count+=1
    else:
        h.append(c[index])
        if index>=0:
            index-=1
#print(h)


for i in range(no_of_bits+no_of_paritybits):
    #print(i)
```

```
i1=list(format(i+1,'b'))
#print(i1)
length=len(i1)-1
#print(length)
if h[i]=='':
    j=length
    m=[]
    for k in range(no_of_bits+no_of_paritybits):
        l=list(format(k+1,'b'))
        k_rev=list(reversed(l))
        if j<len(k_rev):
            if k_rev[j]=='1':
                l1=''.join(l)
                num=int(l1,2)
                m.append(num)
    #print(m)
    num1=0
    for re in range(len(m)):
        #print(i)
        if h[m[re]-1]=='1':
            num1+=1
        else:
            num1==num1
    #print(num1)
    if num1%2==0:
        h[i]='0'
    else:
        h[i]='1'
    #print(h[i])
```

```
result=''.join(h)
result1=''.join(reversed(result))
print("Code transmitted:"+result1)

received=input("Enter the code received:")
received1=''.join(reversed(received))
re=[]
c=[]

for i in received1:
    re.append(i)
for i in range(no_of_bits+no_of_paritybits):
    #print(i)
    i1=list(format(i+1,'b'))
    #print(i1)
    length=len(i1)-1
    #print(length)
    if (i + 1) & i == 0:
        j=length
        m=[]
        for k in range(no_of_bits+no_of_paritybits):
            l=list(format(k+1,'b'))
            k_rev=list(reversed(l))
            if j<len(k_rev):
                if k_rev[j]=='1':
                    l1=''.join(l)
                    num=int(l1,2)
                    m.append(num)
        #print(m)
        num1=0
```

```
    for i2 in range(len(m)):
        #print(i)
        if re[m[i2]-1]=='1':
            num1+=1
        else:
            num1==num1
    #print(num1)
    if num1%2==0:
        c.append('0')
    else:
        c.append('1')
    #print(h[i])

c1=''.join(c)
c2=int(c1,2)
print(f"Error at position:{c2}")
```

Screenshots:

```
>>>
====================== RESTART: C:\Users\vidhi shah\Desktop\python\Sem 5 cn\Exp03-Hamming.py ======================
Enter the code to be transmitted:1011001
Code transmitted:10101001110
Enter the code received:10101101110
Error at position:6
>>>
```

Conclusion:

Thus, we have studied and implemented the CRC and Hamming Code for Error detection and correction.