Exp No. 5

Date: 7.8.25

AIM: Write a program to implement error detection and correction using HAMMING code concept.

Error correction at Data Link Layer:

Hamming code is a set of error-correction codes that can be used to detect and correct the errors that can occur when the data is transmitted from the sender to the receiver.

Sender Program :-

- Apply hamming code concept on the binary data ~~and~~ add redundant bits to it.

```
def hamming_code (data):
    def insert_bits (data):
        m = len (data)
        r = 0
        while (2**r) < (m+r+1):
            r+ = 1
        n = m + r
        result = ['0'] * n
        j = 0
```

```
            for i in range (1, n+1):
                if i & (i - 1)==0 :
                    continue
                result [-i] = ~~bits~~ data[-(j+1)]
                j+ = 1
                if j == m :
                    break
            return result, n, r
    def calc_parity ( pdata , r) :
        n = len ( pdata)
        result = pdata [:]
        for i in range (r) :
            parity_pos =(2 ** i )
            parity_val = 0
            for k in range (1, n+1) :
                if k & parity_pos :
                    parity_val ^= int (result [-k])
            result [-parity_pos] = str( parity_val)
        return result
    pbits, n, r = insert_bits ( data)
    code = calc_parity (pbits, r)
    return ''.join (code)
uinput = input ('Enter binary data :')
print ('Hamming code =', hamming_code (uinput))
```

Receiver Program

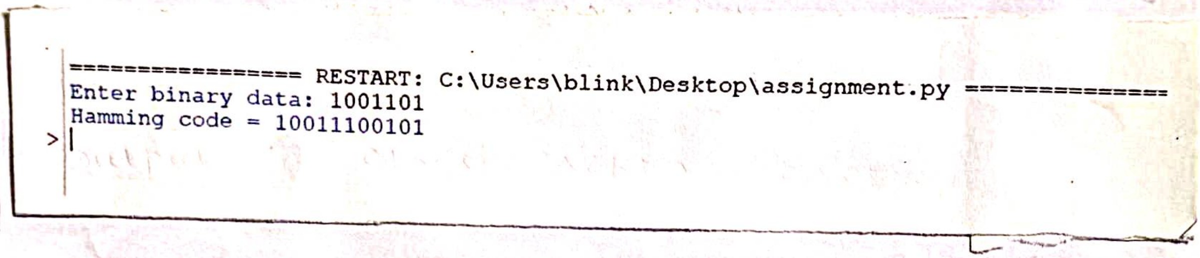
- Apply hamming code on the binary data to check for errors
- If there is any error, display the position of the error

```
def hamming-check (hammingcode):
    n= len (hammingcode)
    r = 0
    while (2**r) < n+1:
        r+=1
    syn = 0
    parity = []
    for i in range(r):
        parity-pos = 2**i
        parity-val = 0
        for k in range (1, n+1):
            if k & parity-pos:
                parity_val ^= int (hammingcode
                                   [-k])
        parity.append (parity-val)
        syn |= (parity_val << i)
    synbits = ''.join (str(x) for x in reversed
                                   (parity))
    return synbits, syn
code = input (''Enter received hamming code:")
res, error = hamming-check (code)
print ('Error bits:', res)
```

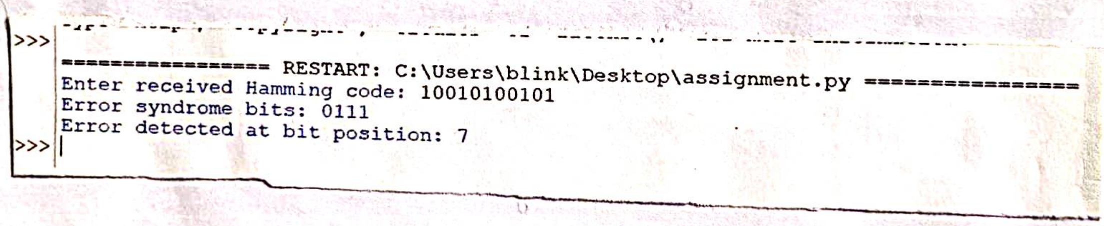
```
if error ==0:
    print ('No error detected')
else :
    print ('Error detected at bit position:',
                                   error)
```

STUDENT OBSERVATION.

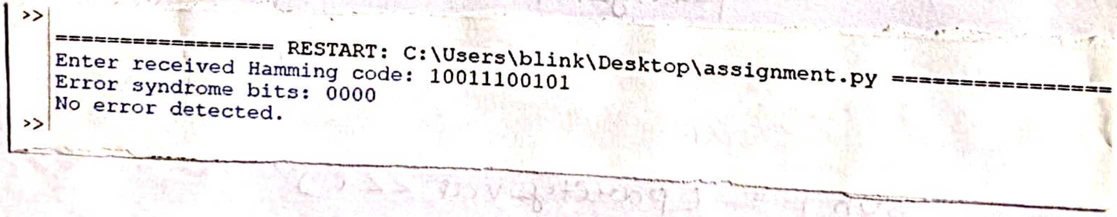
→ Input and Output



```
================= RESTART: C:\Users\blink\Desktop\assignment.py ==============
Enter binary data: 1001101
Hamming code = 10011100101
>|
```



```
>>>
================= RESTART: C:\Users\blink\Desktop\assignment.py ================
Enter received Hamming code: 10010100101
Error syndrome bits: 0111
Error detected at bit position: 7
>>>|
```



```
>>
================= RESTART: C:\Users\blink\Desktop\assignment.py ===============
Enter received Hamming code: 10011100101
Error syndrome bits: 0000
No error detected.
>>
```

RESULT:

Sender and receiver program for hamming code concept was executed and got the output.

8/9/16