7/8/25

PRACTICAL - 6

AIM:

Write a Program to implement error detection and correction using Hamming code concept Make a test run to input data stream and verify error correction feature.

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. It is a technique developed by R.W Hamming for error correction.

Create sender problem with the below feature:

1) Input to sender file should be a text of any length. Program should convert the text to binary.

2) Apply Hamming code concept on the binary data length. Program should convert the text to binary.

3) save this output in a file

Create a receiver program with below features :

1) Receiver program should read the input from channel file.

2) Apply hamming code on the binary data to check error.

3) If there is an error

4) Else remove the redundant bits and convert the binary data to ascii and display the output.

Student Observation

Code:

```
def calc-parity (data):
    C = [0, 0, 0, data[0], 0, data[1],
        data[2], data[3]]
    C[1] = C[3] ^ C[5] ^ C[7]
    C[2] = C[3] ^ C[6] ^ C[7]
    C[4] = C[5] ^ C[6] ^ C[7]

    return C[1]
```

```
def detect - correct (r):

  r[0] +r.
  s1, s2, s4 = r[1] ^ r[3] ^ r[5] ^ r[7],
               r[2] ^ r[3] ^ r[6] ^ r[7],
               r[4] ^ r[5] ^ r[6] ^ r[7]

  def detect - correct (r):

    r = [0] + r
    s1, s2, s4  = r[1] + r[3] ^ r[5] ^ r[2]

    err - pos  =  s4 * 4 + s2 * r + s1
    if err-pos:
       print (f"Error at bit {err-pos}")
         r[err-pos] ^= 1
       print ("Corrected =", r[1:])

     Else:
       print ("No error detected")
    return r[1:]

    def extract_data(c) : return [c[2], c[4], c[5],
                                  c[6]]

    if _name_ - - = "_ _main_ _"
       data = input (" Enter 4-bit data : ")
    if len(data) ! =4 or any (b not in "01" for
                            b in data):
```

```
        exist ("invalid input")
    data = [int (b) for b in data]
    enc = calc_parity (data)
    print ("Encoded:", enc)
   if input ("Introduced error? (y/n):").
                    lower()=='y':
     P= int (input ("Enter position 1-7: "))
     if 1<=p<=7:
        enc[p-1]^=1
     print ("Received :" enc)
     corr= detect_correct (enc)
     print ("original_data:", extra_data
                                    (corr))
```

Output:

```
 Enter :4-bit data: 1011
 Encoded : [0, 1, 1, 0, 0, 1, 1]
 Introducer error : (y/n): y
 Enter position 1-7= 3
 Received : [0, 1, 0, 0, 0, 1, 1]
 Error at bit 3
 Corrected: [0, 1, 0, 1, 0, 1, 1]
 original data [1,0,1,1]
```

Result:

Hence the code for Hamming problem was succesfully executed

26/9/22