Exp No:6
DATE 28.08.25

Hamming code for Errors

AIM:

Write a program to implement error detection and correction using hamming code concept. make a test 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 program:

1) Input to sender file should be a text or any length. program should convert the text to binary.
2) Apply hamming code concept on the binary data and add redundant bits to it.
3. Save the output in a file called channel.

Create a receiver program with below features

1. Receiver program should read the input from channel fire.
2. Apply hamming code on binary data & check error.
3 If there is an error, display the position of

Hamming code for sender:

```
def char to binary:
    return format (ord (ch), '08b')
def hammingi - encode (data 4)
    d1, d2, d3, d4 = (int (bit) for bit in data
        p1 = d1 ^ d2 ^ d4
        p2 = d1 ^ d3 ^ d4
        p4 = d2 ^ d3 ^ d4
    return f" {p1} {p2} {d1} {p4} {d2} {d3
text = input ("Enter text: ")
with open ("channel.text", "w") as f:
        for ch in text:
binch = char. to_binary (ch)
for i in range (0, 8, 4):
    code = hamming - encode (bin_ch(i:i+4
    f.write (code)
print ("Data written to channel.txt with
        Hamming code.")
```

Hamming Code for receiver:

```
def hamming_decode (code 7):
    b = [0] + (int (bit) for bit in code
    p1 = b[1] ^ b[3] ^ b [5] ^ b [7]
    p2 = b[2] ^ b [3] ^ b[6] ^ b[7]
    p4 = b[4] ^ b [5] ^ b [6] ^ b[7]
    error - pos = p1 * 1 + p2 * 2 + p4 * 4
    if error - pos != 0
```

```
print (f " Error detected at Position
{ error-pos }. correcting... ")
    b [error-pos] ^ = 1
  d1, a2, d3, d4 = b[3], b[5], b[6], b[7]
  return f" {d1} {d2} {d3} {d4}"
binary_result = " "
with open ("channel.txt", "r") as f:
    codes = f.read ().
foo i in range (0, len (codes), 7):
    binary_result + = hamming_decode(codes
                          [i : i + 7])
text = " "
for i in range (0, len (binary_result), 8):
    byte = binary_result [i : i + 8].
    text + = chr (int (byte, 2))
Print (" Received text after error Correction
        txt)
```

<u>Input</u>:

Enter 4 bit data : 1011
Sender Side : 0010011
Receiver Side : 0010011.

<u>Output</u>:

Original data bits extracted : 1011

<u>RESULT</u>:

Sender and Reciver Program for Lamming Code concept was executed the output.