Exp NO: 6
Date : 28.8.25

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 corrections feature

Error correction at Data Link Layer:-

Hamming code is a set if 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 reciever. It is a technique developed by R.W Hamming for error correction.

Create sender program:

1. Input to sender file should be a task of any length.
2. Apply hamming code concept on the binary data and add redudant bits to it
3. Save the output in a file called channel.

Create a reciever program with below features:-

1. Receiver Program should read the input from channel files.

2. Apply hamming code on binary data + check errors.

3. If there is an error, display the position of the error.

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

Hamming code for sender :-

```
def char_to_binary :
    return format (ord(ch), '08b')

def hamming_encode (data 4)
    d1, d2, d3, d4 = [int (bit) for the bit in
                      data 4]

    P1 = d1 ^ d2 ^ d4
    P2 = d1 ^ d3 ^ d4
    P4 = d2 ^ d3 ^ d4

    return if "{P1} {P2} {d1} {P4} {d2}
                {d3} {d4}

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])
    b.write(code)
print("Data written to channel.txt with
    hamming code")
```

Hamming code for reciever :-

```
def hamming_decode(code1):
    b = [0] + [int(bit) for bit in code1]
    p1 = b[1] ^ b[3] ^ b[5] ^ b[7]
    p2 = b[2] ^ [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
    printf("Errordetected at position
{error_pos}. correcting.....")

        b[error_pos] ^= 1

    d1, d2, d3, d4 = b[3], b[5], b[6], b[7]
    return f"{d1}{d2}{d3}{d4}"
binary_result = ""
with open("channel.txt", "r") as if
```

```
    code = f.read();
for i in range(0, len(codes), 7):
    binary_result += hamming_
                     decode(codes[i:i+7]]

text = ""
for i in range(0, bn(binary_result,
                    8):
    byte = binary_result[i:i+8];
    text += chr(int(byte, 2))

print("Recieved text after error
correction", txt)
```

Input

Enter 4 bits data : 1101

Sender side : 0010011

Reciever side : 0010011

Output

Original data is bits extracted : 1011

Result

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