Exp: 6

Dt: 09.25

Hamming code

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 that can be used to detect and correct the errors that can occur when data is transmitted from sender to the receiver. It is a technique developed by R. W. Hamming for error correction.

create sender program with below features.

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 binary data and add redundant bits to it.
3. Save this output in a file called channel.

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 for errors.

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

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

Student observation:

```
# Hamming code sender
def char_to_binary(ch):
    return format(ord(ch), '08b')
def hamming_encode(data4):
    d1, d2, d3, d4 = [int(bit) for bit in data4]
    P1 = d1 ^ d2 ^ d4
    P2 = d1 ^ d3 ^ d4
    P4 = d2 ^ d3 ^ d4
    return f"{P1}{P2}{d1}{P4}{d2}{d3}{d4}"
text = input("Enter text: ")
with open("channel.txt", "w") as f:
```

```
for ch in text:
    bin_ch = 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 Receiver
def hamming_decode(code):
    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, 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 f:
    codes = f.read()
```

```
for i in range(0, len(code), 7):
    byte = binary_result[i:i+8]
    text += chr(int(byte, 2))
print("Received text after error correction:", text)
```

Input:

Enter text to send : hi

output:

Data written to channel : 1001101 1100111...

Decoded text at receiver : hi

Result:

Hence the hamming code is implemented to detect and word errors.