Ex.No: 6

Practical-6

Aim: Write a program to implement error detection and correction using Hamming Code concept.

Error correction with Hamming code

Sender program:

- Take text input
- Convert text -> binary
- Apply hamming code (add redundant bits)
- Save output to channel file.

Receiver program:

- Read data from channel file
- Check errors using Hamming code.
- If error -> show error position.
- If no error -> remove redundant bits.
- Convert binary -> ASCII, display text.

Program:

```
def main():
    data = list(map(int, input("Enter 4 databits
                       (eg., 1011)", "").split()))

    d1, d2, d3, d4 = data
    P1 = d1^d2^d4
    P2 = d1^d3^d4
    P3 = d2^d3^d4

  code = [P1, P2, d1, P3, d2, d3, d4)
  Print ("Encoded Hamming code.", "".join(mapcode))
```

```
recv = list(map(int, input("Enter received 7 bits: ")split()))
    c1 = recv[0] ^ recv[2] ^ recv[4] ^ recv[6]
    c2 = recv[1] ^ recv[2] ^ recv[5] ^ recv[6]
    c3 = recv[3] ^ recv[4] ^ recv[5] ^ recv[6]
  error pos = c1 + (c2<<1) + (c3 <<2)
 if error pos = 0:
      print("No error detected")
 else:
    print("error at bit position: ", error Pos)
    recv[error POS -1]^ = 1
    print("corrected code: ", "" join(map(str, recv)))
  if_name_ == "_main_":
      main()
```

Sample Inputs Outputs:

```
Enter 4 data bits: 1011
Encoded Hamming code: 0110011
Enter received 7 bits: 0111011
Error at bit position: 4
Corrected code: 0110011
```

Result:

Hence the required program the error detection & error correction is written & executed successfully.