Practical 13

Error Correction : Hamming Code

* Objective

To implement Hamming Code for error detection and correction in data transmission, allowing the user to input a data word, calculate its corresponding Hamming Code, and identify any single-bit errors in the received code word.

* Code

```
#include <stdio.h>
#include <math.h>
int input[32];
int code[32];
int ham_calc(int,int);
void main()
{
   int n,i, p_n=0, c_l, j, k;
   printf("Please enter the length of the
            Data Word: ");
   scanf("%d", &n);
   printf("Please enter the Data Word:\n");
   for(i=0; i<n; i++)
   {
      scanf("%d", &input[i]);
   }
   i = 0;
   while(n > (int)pow(2,i) - (i+1))
   {
```

```
        p_n++;
        i++;
    }

    c_l = p_n + n;

    j = k = 0;
    for (i = 0; i < c_l; i++)
    {
        if (i == ((int)pow(2,k) - 1))
        {
            code[i] = 0;
            k++;
        }
        else
        {
            code[i] = input[j];
            j++;
        }
    }
    for (i=0; i < p_n; i++)
    {
        int position = (int)pow(2,i);
        int value = hum_calc(position, c_l);
        code[position - 1] = value;
    }
    printf("\n The calculated Code Word is: ");
    for (i = 0; i < c_l; i++)
        printf("%d", code[i]);
    printf("\n");
    printf("Please enter the received Code
            Word : \n");
```

```
for (i = 0; i < c_l; i++)
    scanf("%d", &code[i]);
int error_pos = 0;
for (i = 0; i < p_n; i++)
{
    int position = (int)pow(2,i);
    int value = ham_calc(position, c_l);
    if (value != 0)
        error_pos += position;
}
if (error_pos == 1)
    printf("The received Code Word is correct.
            \n");
else
    printf("Error at bit position: %d \n",
            error_pos);
}
int ham_calc(int position, int c_l)
{
    int count = 0, i, j;
    i = position - 1;
    while(i < c_l)
    {
        for(j = i; j < i + position; j++)
        {
            if(code[j] == 1)
                count++;
        }
        i = i + 2 * position;
    }
    if(count % 2 == 0)
        return 0;
    else
        return 1;
```

* Output.

```
Please enter the length of the Data Word: 4
Please enter the Data Word:
1 0 1 1

The calculated Code Word is : 0110011
Please enter the received Code Word:
0110011
The received Code Word is correct.
```

* Learning Outcome:

Understand and implement Hamming Code for error detection and correction in digital communication systems.