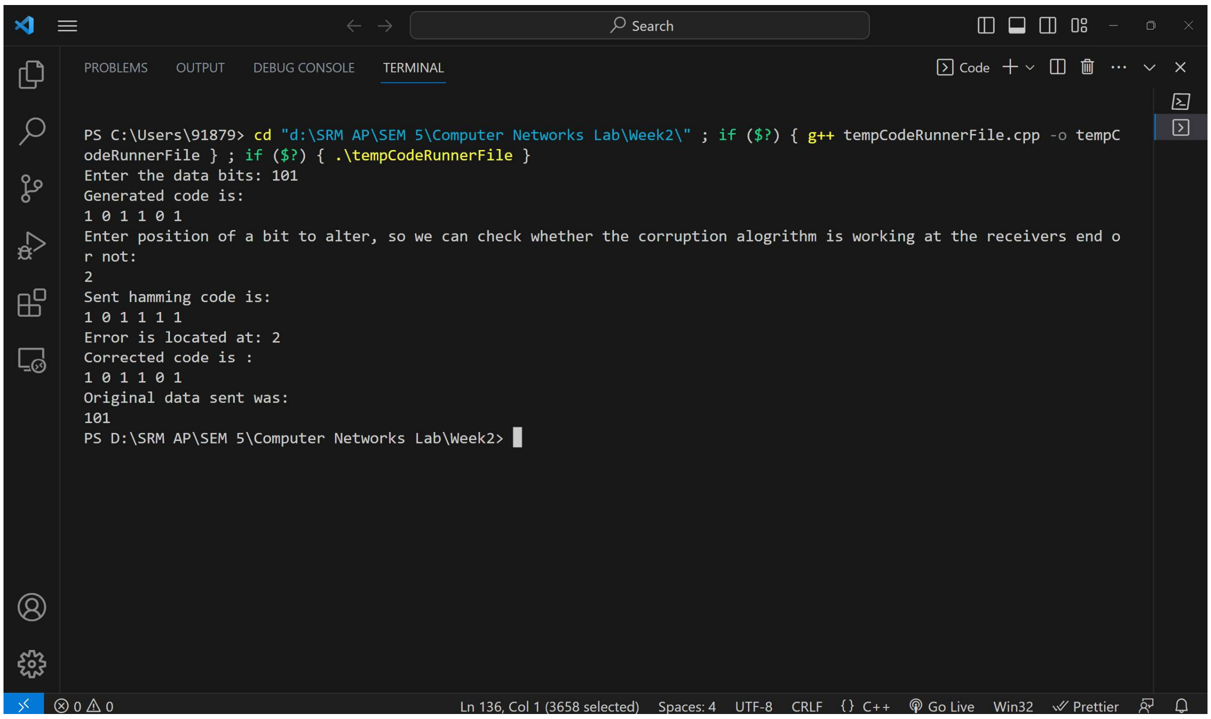
Lab Experiment – 2

1. Write a program to implement Error Detection and Correction Technique using Hamming code.

```cpp
#include <iostream>
#include <cmath>
#include <string>
using namespace std;
bool isInteger(double number)
{
    return floor(number) == number;
}
void display_hamming_code(int *hamming_code, int total_count)
{
    for (int i = total_count - 1; i >= 0; i--)
    {
        cout << hamming_code[i] << " ";
    }
    cout << endl;
}
int *generate_hamming_code(string data, int &total_count)
{
    int data_count = data.length();
    int parity_count = 1;
    while (pow(2, parity_count) < data_count + parity_count + 1)
    {
        parity_count += 1;
    }
    total_count = data_count + parity_count;
    int *hamming_code = new int[total_count];
    int data_ptr = 0;
    for (int i = total_count; i >= 1; i--)
    {
        if (isInteger(log2(i)))
        {
            hamming_code[i - 1] = 2;
        }
        else
        {
            hamming_code[i - 1] = (int)data[data_ptr++] - 48;
        }
    }
    int count;
    for (int i = 1; i <= total_count; i = i * 2)
    {
        count = 0;
        for (int j = i + 1; j <= total_count; j++)
        {
            if (j & (1 << (int)(log2(i))))
            {
                if (hamming_code[j - 1])
                {
                    count++;
                }
            }
        }
        if (count % 2 == 0)
        {
            hamming_code[i - 1] = 0;
        }
        else
        {
            hamming_code[i - 1] = 1;
        }
    }
    return hamming_code;
}
int check(int *hamming_code, int total_count)
{
    string bit_info = "";
    int count;
    for (int i = 1; i <= total_count; i = i * 2)
    {
        count = 0;
        for (int j = i + 1; j <= total_count; j++)
        {
            if (j & 1 << (int)(log2(i)))
            {
                if (hamming_code[j - 1])
                {
                    count++;
                }
            }
        }
        if (hamming_code[i - 1])
        {
            count++;
        }
        if (count % 2 == 0)
        {
            bit_info = "0" + bit_info;
        }
        else
        {
            bit_info = "1" + bit_info;
        }
    }
    return stoi(bit_info, 0, 2);
}
string retrive_data_from_hammingcode(int *hamming_code, int total_count)
{
    string retrived_data = "";
    for (int i = total_count; i >= 1; i--)
    {
        if (isInteger(log2(i)))
        {
            continue;
        }
        else
        {
            retrived_data += to_string(hamming_code[i - 1]);
        }
    }
    return retrived_data;
}
int main()
{
    string data;
    cout << "Enter the data bits: ";
    cin >> data;
    int total_count;
    int *hamming_code = generate_hamming_code(data, total_count);
    cout << "Generated code is: " << endl;
    display_hamming_code(hamming_code, total_count);
    cout << "Enter position of a bit to alter, so we can check whether the
corruption alogrithm is working at the receivers end or not: " << endl;
    int bit;
    cin >> bit;
    hamming_code[bit - 1] = hamming_code[bit - 1] ^ 1;
    cout << "Sent hamming code is: " << endl;
    display_hamming_code(hamming_code, total_count);
    int corrupted_bit = check(hamming_code, total_count);
    cout << "Error is located at: " << corrupted_bit << endl;
    hamming_code[corrupted_bit - 1] = hamming_code[corrupted_bit - 1] ^ 1;
    cout << "Corrected code is : " << endl;
    display_hamming_code(hamming_code, total_count);
    cout << "Original data sent was: " << endl;
    string retrived_data = retrive_data_from_hammingcode(hamming_code,
total_count);
    cout << retrived_data;
}
```

Output:



```
PS C:\Users\91879> cd "d:\SRM AP\SEM 5\Computer Networks Lab\Week2\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempC
odeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter the data bits: 101
Generated code is:
1 0 1 1 0 1
Enter position of a bit to alter, so we can check whether the corruption alogrithm is working at the receivers end o
r not:
2
Sent hamming code is:
1 0 1 1 1 1
Error is located at: 2
Corrected code is :
1 0 1 1 0 1
Original data sent was:
101
PS D:\SRM AP\SEM 5\Computer Networks Lab\Week2>
```

Ln 136, Col 1 (3658 selected)   Spaces: 4   UTF-8   CRLF   {} C++   Go Live   Win32   Prettier