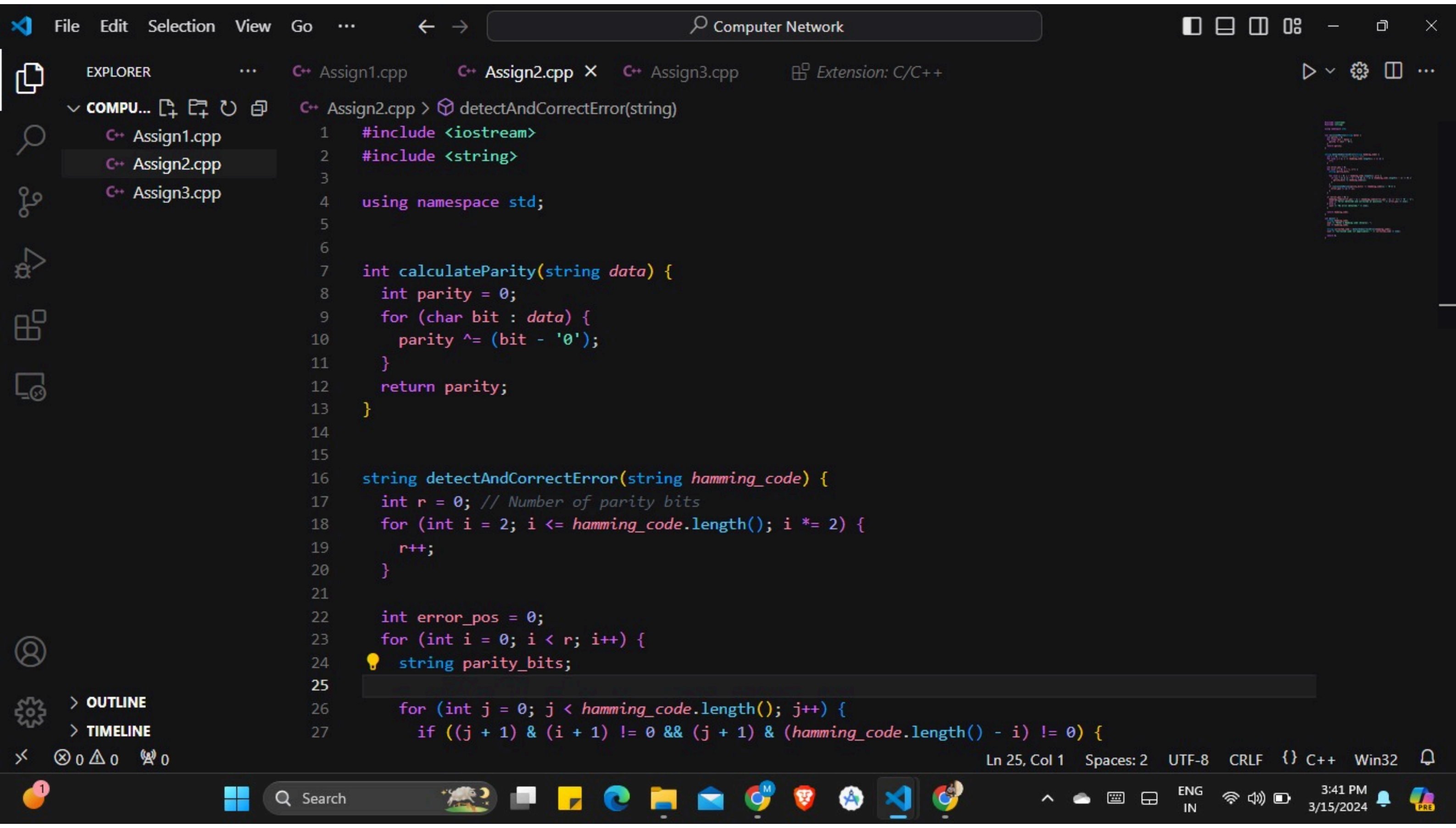
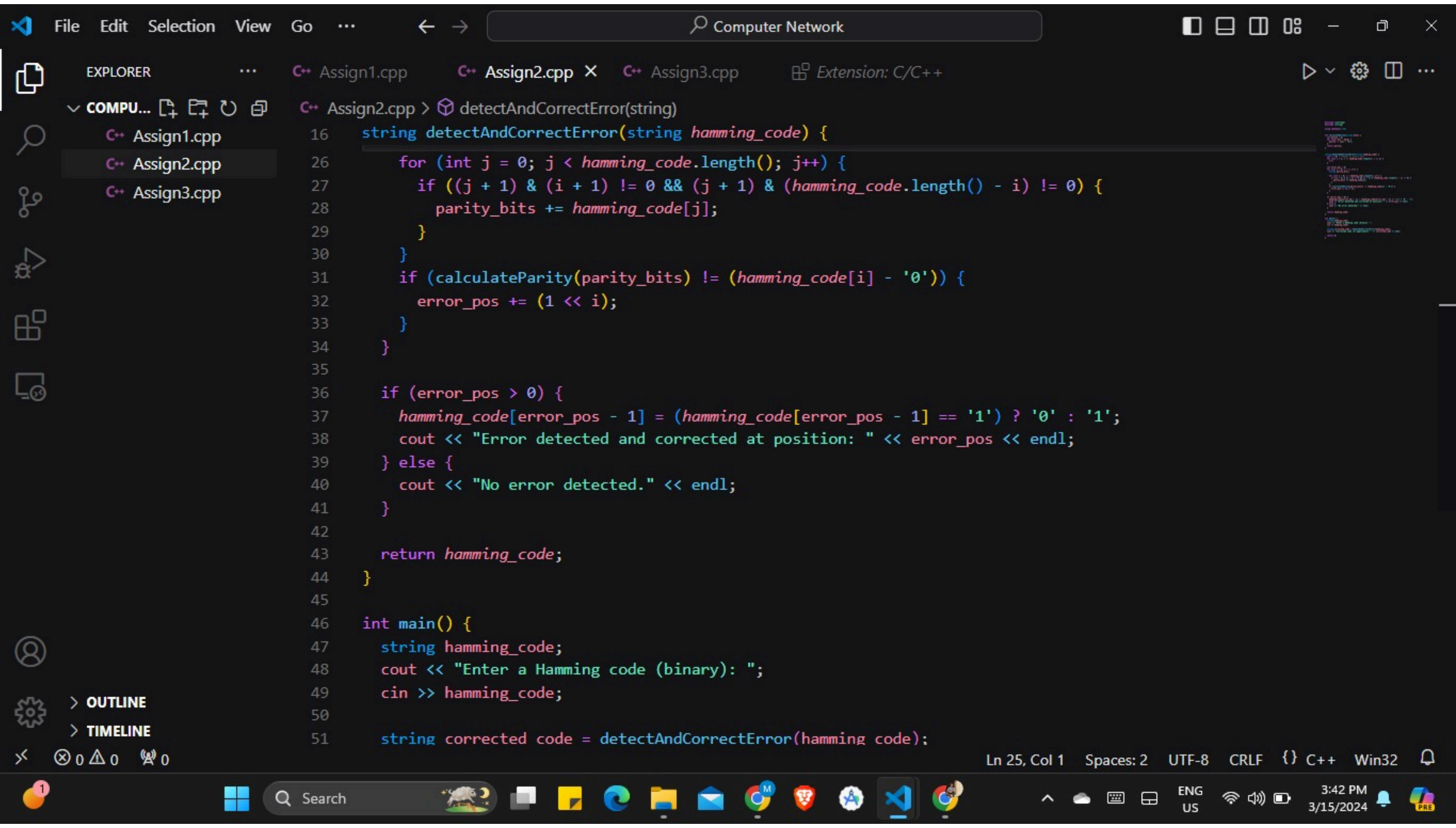
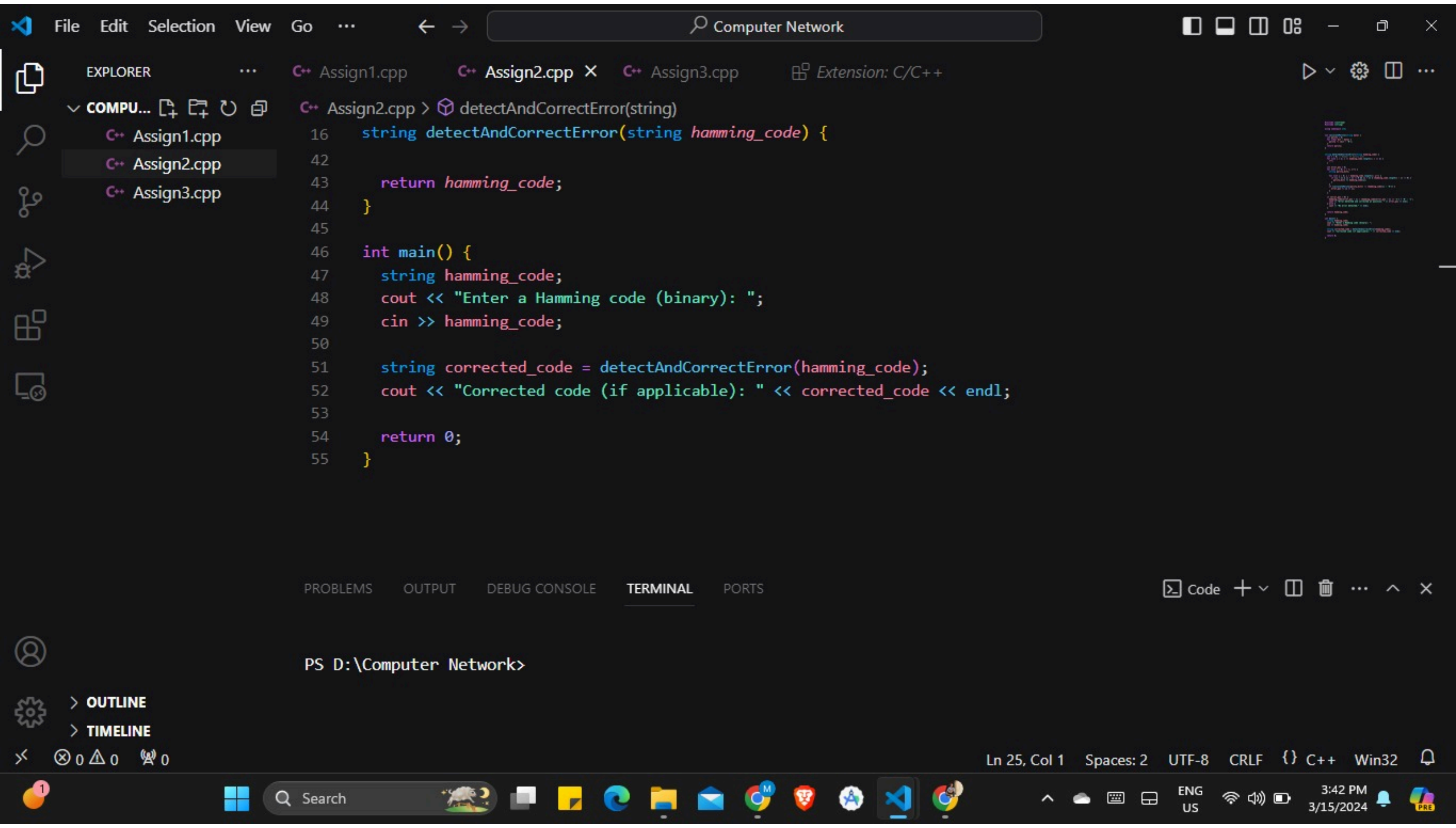


```cpp
#include <iostream>
#include <string>

using namespace std;


int calculateParity(string data) {
  int parity = 0;
  for (char bit : data) {
    parity ^= (bit - '0');
  }
  return parity;
}


string detectAndCorrectError(string hamming_code) {
  int r = 0; // Number of parity bits
  for (int i = 2; i <= hamming_code.length(); i *= 2) {
    r++;
  }

  int error_pos = 0;
  for (int i = 0; i < r; i++) {
    string parity_bits;

    for (int j = 0; j < hamming_code.length(); j++) {
      if ((j + 1) & (i + 1) != 0 && (j + 1) & (hamming_code.length() - i) != 0) {
```



```cpp
string detectAndCorrectError(string hamming_code) {
    for (int j = 0; j < hamming_code.length(); j++) {
      if ((j + 1) & (i + 1) != 0 && (j + 1) & (hamming_code.length() - i) != 0) {
        parity_bits += hamming_code[j];
      }
    }
    if (calculateParity(parity_bits) != (hamming_code[i] - '0')) {
      error_pos += (1 << i);
    }
  }

  if (error_pos > 0) {
    hamming_code[error_pos - 1] = (hamming_code[error_pos - 1] == '1') ? '0' : '1';
    cout << "Error detected and corrected at position: " << error_pos << endl;
  } else {
    cout << "No error detected." << endl;
  }

  return hamming_code;
}

int main() {
  string hamming_code;
  cout << "Enter a Hamming code (binary): ";
  cin >> hamming_code;

  string corrected_code = detectAndCorrectError(hamming_code);
```



```cpp
string detectAndCorrectError(string hamming_code) {

  return hamming_code;
}

int main() {
  string hamming_code;
  cout << "Enter a Hamming code (binary): ";
  cin >> hamming_code;

  string corrected_code = detectAndCorrectError(hamming_code);
  cout << "Corrected code (if applicable): " << corrected_code << endl;

  return 0;
}
```

```
PS D:\Computer Network>
```