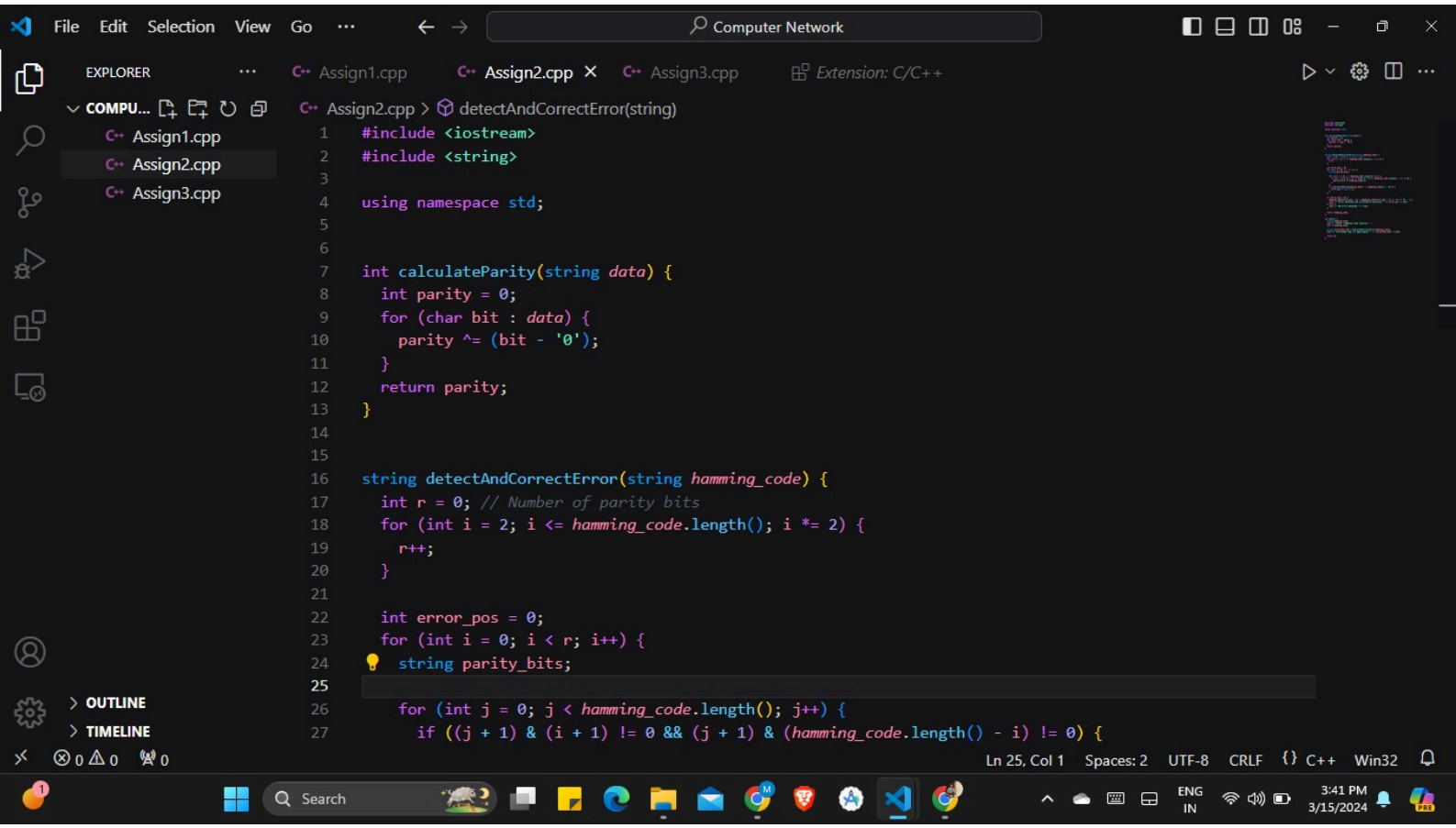
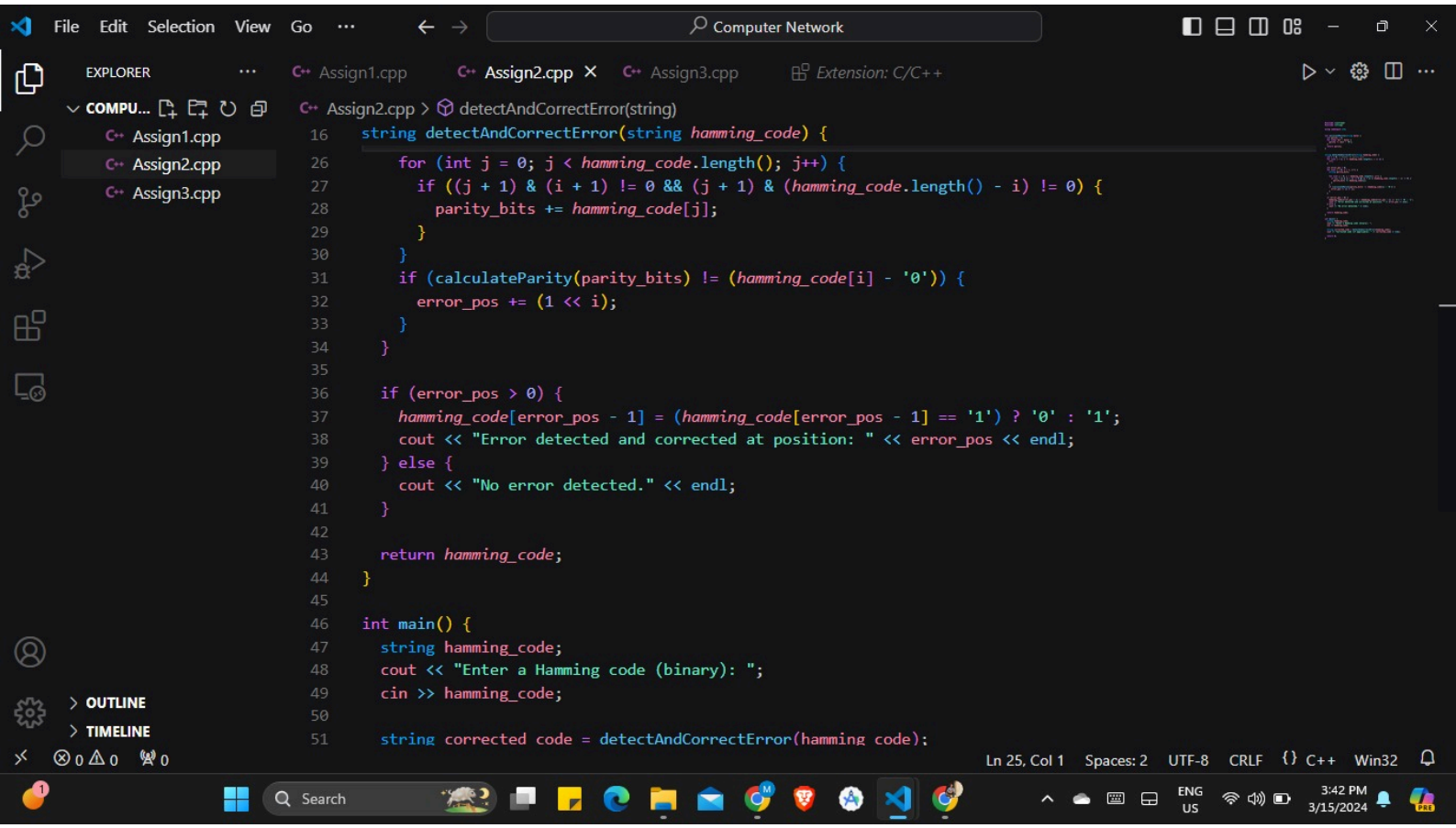
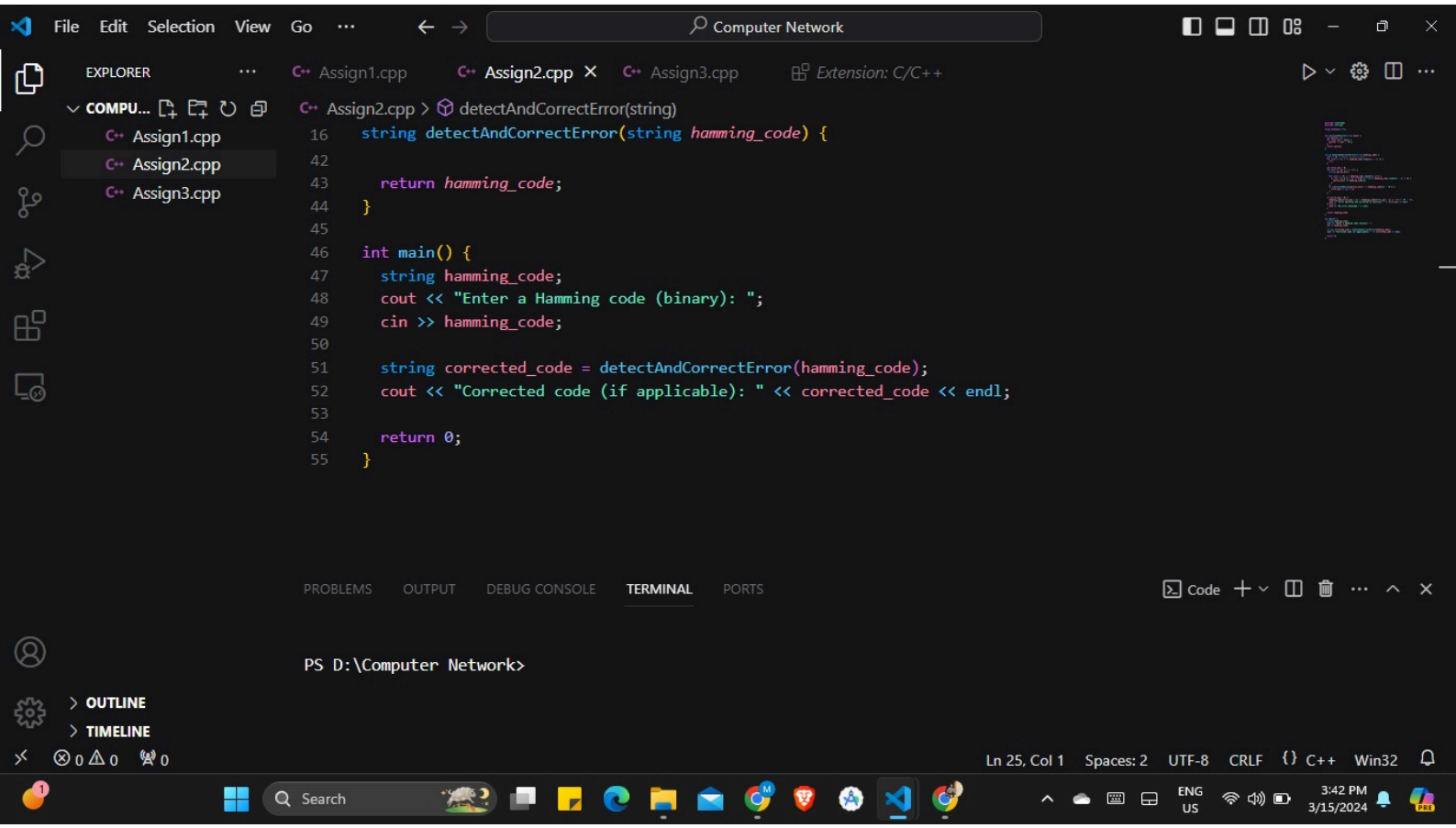

File Edit Selection View Go
Computer Network
EXPLORER
COMPU...
Assign1.cpp
Assign2.cpp
Assign3.cpp
Assign1.cpp
Assign2.cpp
Assign3.cpp
Extension: C/C++
Assign2.cpp > detectAndCorrectError(string)
#include <iostream>
#include <string>
using namespace std;
int calculateParity(string data) {
int parity = 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) {
OUTLINE
TIMELINE
Ln 25, Col 1
Spaces: 2
UTF-8
CRLF
C++
Win32
Search
ENG
IN
3:41 PM
3/15/2024


File Edit Selection View Go
Computer Network
EXPLORER
Assign1.cpp
Assign2.cpp
Assign3.cpp
Extension: C/C++
Assign2.cpp > detectAndCorrectError(string)
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);
OUTLINE
TIMELINE
Ln 25, Col 1 Spaces: 2 UTF-8 CRLF C++ Win32
Search
ENG US
3:42 PM 3/15/2024


File Edit Selection View Go
Computer Network
EXPLORER
COMPU...
Assign1.cpp
Assign2.cpp
Assign3.cpp
Assign1.cpp
Assign2.cpp
Assign3.cpp
Extension: C/C++
Assign2.cpp > detectAndCorrectError(string)
16 string detectAndCorrectError(string hamming_code) {
42
43 return hamming_code;
44 }
45
46 int main() {
47 string hamming_code;
48 cout << "Enter a Hamming code (binary): ";
49 cin >> hamming_code;
50
51 string corrected_code = detectAndCorrectError(hamming_code);
52 cout << "Corrected code (if applicable): " << corrected_code << endl;
53
54 return 0;
55 }
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
PS D:\Computer Network>
OUTLINE
TIMELINE
Ln 25, Col 1
Spaces: 2
UTF-8
CRLF
C++
Win32
Search
ENG
US
3:42 PM
3/15/2024