

FIC - CMPT 125 202402 - Assignment 1

Due Date and Time: Monday 3 June 2024 at 11:55PM PST

Instructor: Dr. Yonas T. Weldeselassie (Ph.D.)

Read this document in its entirety and carefully before you start to work on the assignment and understand it. If you have any questions, don't hesitate to email me.

In this assignment, we will work on numeric information representation including unsigned binary representation and two's complement binary representation. In addition, we will demonstrate arithmetic of integers in two's complement binary arithmetic.

In order to store the different binary representations of integers, we will use the C++ string data type. You are not allowed to use any ARRAY variable in this assignment. Read the restriction section below.

Consider the program given below.

```
#include <iostream>
#include <cmath>
#include <string>
using namespace std;

string getUnsignedBinary(int x, int L)
{
    //Pre-condition: x >= 0 and L > 0
    //Post-condition: the unsigned binary representation of x in L bit pattern is returned as a C++ string data type
    //Any overflow bits are ignored

    //PUT YOUR CODE HERE
}
int getUnsignedDecimal(string s)
{
    //Pre-condition: s is the unsigned binary representation of a decimal number
    //Post-condition: The unsigned decimal number value of s is returned

    //PUT YOUR CODE HERE
}
string getBinarySum(string A, string B)
{
    //Pre-condition: A and B are non-empty, have equal lengths, and store '0's and '1's characters
    //Post-condition: The binary sum of A and B is returned as a C++ string data type with the same length as A and B
    //Any overflow bits are ignored

    //PUT YOUR CODE HERE
}
string getTwosComplementBinary(int x, int L)
{
    //Pre-condition: x is a signed integer and L > 0
    //Post-condition: the two's complement binary representation of x in L bit pattern is returned as a C++ string
data type
    //Any overflow bits are ignored

    //PUT YOUR CODE HERE
}
int getSignedDecimal(string s)
{
    //Pre-condition: s is the twos complement binary representation of a decimal number
    //Post-condition: The signed decimal number value of s is returned

    //PUT YOUR CODE HERE
}
int selectComputation()
{
    cout << "Select your computation" << endl;
    cout << "    1. Unsigned Binary Representation Computation" << endl;
```

```

cout << " 2. Two's Complement Representation Computation" << endl;
cout << " 3. Exit Program" << endl;
int selection;
cout << "Enter your selection (1, 2, or 3): ";
cin >> selection;
while (selection != 1 && selection != 2 && selection != 3)
{
    cout << "Please enter a correct choice: ";
    cin >> selection;
}
return selection;
}
int main()
{
    cout << "This program demonstrates numeric information representation including" << endl;
    cout << " *** Unsigned Binary Representation," << endl;
    cout << " *** Conversion from unsigned binary to decimal," << endl;
    cout << " *** Two's Complement Binary Representation," << endl;
    cout << " *** Two's complement binary addition, and" << endl;
    cout << " *** Conversion from two's complement to decimal." << endl << endl;
    do
    {
        int selection = selectComputation();
        if (selection == 1)
        {
            int bit_pattern_size, num;
            cout << endl << "Enter a positive integer for the bit pattern size: ";
            cin >> bit_pattern_size;
            while (bit_pattern_size <= 0)
            {
                cout << "You must enter a positive integer. Enter again please: ";
                cin >> bit_pattern_size;
            }
            cout << "Enter a non-negative integer: ";
            cin >> num;
            while (num < 0)
            {
                cout << "You must enter a non-negative integer. Enter again please: ";
                cin >> num;
            }
            string s = getUnsignedBinary(num, bit_pattern_size);
            cout << "The unsigned binary representation of " << num << " in " << bit_pattern_size << " bit is " << s
<< endl;
            int value = getUnsignedDecimal(s);
            cout << "This unsigned binary represents the decimal number " << value << endl;
            if (value == num)
                cout << "This is a correct result." << endl;
            else
                cout << "This is not correct result because our bit pattern is too small to store the given decimal
number." << endl;
                cout << endl;
            }
            else if (selection == 2)
            {
                int bit_pattern_size, num1, num2;
                cout << endl << "Enter a positive integer for the bit pattern size: ";
                cin >> bit_pattern_size;
                while (bit_pattern_size <= 0)
                {
                    cout << "You must enter a positive integer. Enter again please: ";
                    cin >> bit_pattern_size;
                }
                cout << "Enter an integer: ";
                cin >> num1;
                string s1 = getTwosComplementBinary(num1, bit_pattern_size);
                cout << "The two's complement binary representation of " << num1 << " in " << bit_pattern_size << " bit is
" << s1 << endl;
                cout << endl;
                cout << "Enter a second integer: ";
                cin >> num2;
                string s2 = getTwosComplementBinary(num2, bit_pattern_size);
                cout << "The two's complement binary representation of " << num2 << " in " << bit_pattern_size << " bit is
" << s2 << endl;

```

```

        cout << endl;
        string s3 = getBinarySum(s1, s2);
        cout << "The binary sum of " << s1 << " and " << s2 << " is " << s3 << endl;
        int sum = getSignedDecimal(s3);
        cout << "The sum of " << num1 << " and " << num2 << " is " << num1+num2 << " and the integer value of the
binary sum is " << sum << endl;
        if (sum == num1 + num2)
            cout << "This is a correct result." << endl;
        else
            cout << "This is not correct result because our bit pattern is too small to store the sum." << endl;
    }
    else
        break;
    system("Pause");
    cout << endl << endl;
}while (true);

system("Pause");
return 0;
}

```

Sample Run Output

You are provided a text file named Starter Code.txt that is uploaded together with this assignment. You are provided lots of sample run outputs in this text file to demonstrate the program. Use the sample run outputs to test your work.

Your Task

You are required to implement the functions needed by the program whose declarations and whose pre and post conditions are already provided for you as shown above. These functions should be defined so that the program given above runs without any syntax, runtime, or semantic errors.

In order to easily copy and paste the code given above, the Starter Code.txt file also contains the same code which you need to start your work with.

Restrictions and Requirements

- You are required to use C++ string Data Type to represent the two's complement binary representations.
- You may define extra supporting/helper functions if you wish.
- You are not allowed to add any include directive to the program.
- You are not allowed to use any ARRAY variable for any purpose.
- If you use ARRAY variable for any purpose in your program, you will automatically get zero mark.

Submission Format

You will find a submission link for Assignment 1 on Moodle under Week 3 and you are required to submit through Moodle your source code file (that is .cpp file) that must contain the given program and all the functions definitions. You must also include in your submission any extra supporting/helper function definition if you have defined any.

Submission Due Date and Time

The due date and time to submit your assignment work online through Moodle is Monday **3 June 2024 at 11:55PM**. Moodle will not allow you to submit after the due date and time. Moreover no submission through email or printed copy will be accepted. No submission on time through Moodle means receiving zero marks for the assignment. You are therefore strongly advised to submit before the due date and time.

How to Submit Your Assignment Work

For demonstration purposes, assume that your project is saved as **Desktop\CMPT125\Assignment1**. In order to submit your assignment, you will need to submit your source code file that you will find at **Desktop\CMPT125\Assignment1\Assignment1\Assignment1.cpp** as follows

- **Step 1.** Click on the **Submission Link** on Moodle
- **Step 2.** Click on **Add submission** button
- **Step 3.** Drag your program source code file found at **Desktop\CMPT125\Assignment1\Assignment1\Assignment1.cpp** to the submission box and you are done.

If you don't know how to drag your file to the submission box shown, then you may alternatively submit it as follows:

- **Step 1.** Click the submission icon as shown in the diagram below



- **Step 2.** You will see a new window. Click **Browse** button
- **Step 3.** Select your file located at **Desktop\CMPT125\Assignment1\Assignment1\Assignment1.cpp**
- **Step 4.** Click the **Open** button
- **Step 5.** Click the **Upload this file** button
- **Step 6.** Click the **Save changes** button

This completes the submission process.

Updating (Modifying) Your Submission

You may also update (modify) your submission before the due date and time by first removing it and then re-submitting again. In order to re-submit again, do the following:

- **Step 1.** Click on the **Edit submission** button
- **Step 2.** Right click on the submitted file and click on the **Delete** button
- **Step 3.** Submit again by following the steps described above

This completes your uploading/updating your assignment work.