**CS 270: Assignment 02**

**Due:** Wednesday, February 22, 2017, at 11:59 pm (Central)

# Data Representation in Java

In this assignment you will be implementing procedures to convert decimal numbers into two’s complement binary numbers and to evaluate binary arithmetic. While there are libraries available to help you do this, you are **NOT** allowed to use them.

As we discussed in class, positional number systems encode numbers as a sequence of digits. In Java, we can use an array of char to represent these sequences, and hence numbers in various positional number systems. In the decimal number system, valid sequences consist of the digits 0 through 9, optionally preceded by either a + or − to indicate the sign; in binary, valid sequences contain only the digits 0 and 1. In your program, you will need to represent binary numbers using char arrays.

**Template code** is available on D2L to help you get started. This includes a driver class (Driver) for handling input and output, and a solver class (Solver) which contains several method stubs that you will need to implement in the course of this assignment. You may write your own helper methods in the Solver, but you should **only** use native Java structures; do **not** import or use any additional Java libraries or classes (e.g., Math, Character and String).

# Assignment Details

The Solver class is described below:

public Solver()

Constructor which instantiates and initializes an array of 16 integers named powersOf2, which is a private class member, with the first 16 powers of 2 (from 20 through 215).

public int howManyBits(int value)

Determines the *minimal* number of bits that are needed to represent the given decimal value as a two’s complement binary number. You may assume that inputs will fit in 16 bits.

public char[] decimalToBinary(int value)

Converts the given decimal value into two’s complement binary representation using the *minimal* number of bits necessary. (That is, the size of the returned array should depend on the input value.)

public char[] twosComplementNegate(char[] binaryStr)

Returns the negation of the given two’s complement binary number.

public char[] signExtend(char[] binaryStr, int numBits)

Performs sign extension on the given two’s complement binary number to fit in the given number of bits.

public char[] evaluateExpression(char[] binaryStr1, String op, char[] binaryStr2) Evaluates the given arithmetic expression involving two’s complement binary numbers and returns the result. This method should do the following:

* If the numbers of bits used to represent the two numbers are not equal, sign-extend the number with fewer bits so that they are equal.

CS 270 **Assignment 02** Page 2 of 2

* Perform the appropriate arithmetic operation on the two binary numbers and return the result. If there is overflow, an asterisk (\*) should be added at the beginning of the returned array.

You should implement the basic addition algorithm for binary numbers when writing this method. Do **not** convert the numbers to decimal in order to evaluate the expression.

I recommend writing the methods in the order shown and testing thoroughly before moving from one method to the next. It may also be helpful to test with small, positive numbers first, before implementing and testing with negative numbers.

Some example runs of the program are shown below (user input is colored). Note that the expression consists of two signed decimal numbers (int) and an operator (String). There should be at least one space between each number and the operator.

Example 1:

Enter an expression: 5 + 1

5 + 1 evaluates to the following in binary: 0110

Example 2:

Enter an expression: 42 + 10

42 + 10 evaluates to the following in binary: 0110100

Example 3:

Enter an expression: -7 - 2

-7 - 2 evaluates to the following in binary: \*0111

# Submission Details

This is an **individual** assignment. You should submit a **single** .java **file** of your source code (the Solver class) to the Assignment 2 submission site on **Autolab and D2L**. Your code should adhere to the Java style guidelines posted to D2L, and it should compile and run without errors. Code that does not compile will receive a **significant** penalty.

More information on how to get an account on Autolab and how to submit work is available on

Piazza.

# Grading Guidelines

You will be assessed on the correctness of your code, the quality of your solutions, and adherence to the style guidelines. This assignment is worth 100 points in the Assignments grade category.