Lab #1 CoSc 20203 Spring 2023

Xerox Sigma 9 Machine Instructions

Due: 16 Feb

PROBLEM

The Xerox Sigma series of mainframe computers was announced in 1970 to compete with IBM's System/360 series. There were many instructions with multiple formats. We will consider only four instructions. An instruction in assembly language is written like

```
LW,5  1000,7
AW,5  *1080
```

The meaning of an instruction is not important for this assignment, but to partially satisfy your curiosity the first instruction says to load a value into register 5 from the memory location specified by 1000 plus the contents of register 7. The second instruction says to add the contents of the location specified by the contents of location 1080 to the contents of register 5.

In machine language, the instructions above look like

```
Binary                                  Hex
00110010 01011110 00000011 11101000     32 5E 03 E8
10110000 01010000 00000100 00111000     B0 50 04 38
```

A blank has been inserted between bytes to improve readability.

The format of the instructions that we will consider is

```
Assembly Language       Machine Language
LI,R   V                00100010 rrrrvvvv vvvvvvvv vvvvvvvv
LW,R   [*]D[,X]         *0110010 rrrrxxxd dddddddd dddddddd
AW,R   [*]D[,X]         *0110000 rrrrxxxd dddddddd dddddddd
STW,R  [*]D[,X]         *0110101 rrrrxxxd dddddddd dddddddd
```

Where “R” is a register (0 – 15)
"X" is an index register (0 – 7)
"V" is a value (-524288 – 524287)
“D” is the displacement 0 – 131071)
and […] is optional

If the asterisk is present in assembly, then the high-order bit is 1, otherwise it is 0.
If X is not specified, then those bits are zero.
The format must be exactly as shown although the number of blanks between the first part of the instruction and the second is not important.
In the assembly language, all numbers are decimal.

ASSIGNMENT

You will be given a Java application with a graphical user interface that allows a user to encode and decode Sigma 9 instructions. The user will enter a hex number or binary number and request that it be decoded into assembly language or enter an assembly language instruction and request that it be encoded into hex and binary representations.

You must write the code for the actionPerformed methods for the "Encode" and "Decode" buttons. Your program needs to encode/decode only the four instructions shown above.

Use the errorLabel TextField to display an error message if the input cannot be encoded/decoded or there are formatting errors. You should use my reference implementation for examples. Remember that users are careless. Be thorough in testing. My grader will deduct points if you encode or decode something that is not valid.

NOTE:

1. I included methods for converting an int to a String of binary digits and a String of hex digits. Please use them. If you use the corresponding methods in the standard Java distribution, you will not get the right answer.
2. To convert a String of binary digits or a String of hex digits, use the Long.parseLong method and then cast the result to an int. If you use the Integer.parseInt method, you will not get the right answer for some instructions.
3. DO NOT CHANGE ANYHING THAT I WROTE. You can add code but you cannot change anything. My grader will deduct points, in particular if you change the name of the class.
4. This lab is intended to give you experience in using bitwise operators. Do NOT use strings to create or manipulate the machine language instruction.
   a) To decode a machine language instruction, you must convert the input to an int and then use the bitwise operators to isolate each piece that corresponds to an element of the assembly language instruction.
   b) To encode an assembly language instruction, you must start with a zero-valued int and then use the bitwise operators to set the relevant bits for each element of the assembly language instruction.

If you find an error in my reference implementation and you are the first to notify me, you get 5 points extra credit on the assignment.