ELE209 Lab 6b
LC-3 Assembly Language Programming and Debugging

Due: You must email your code to the TA and demonstrate it by the **start** of lab the next week. Make sure your code is an emailed as an attachement, and if you worked with someone else in lab, be sure to identify your lab partner in the email.

Style requirements: You must follow the style guidelines given at the end of last weeks assignment, and **20% of your grade will be dependent on following these guidelines.**

This is the second part of a two week lab. The subtasks must be completed and demonstrated to the TA in the order they are given. Subtasks 1 and 2 were completed in the previous lab.

Assignment: You will write an assembly language program that will convert a number entered from the keyboard into a 16 bit two's compliment value, then display it to the console as a binary number.

The numbers have a form similar to how you enter values into registers in the LC-3 simulator. Specifically, they have the form

```
[SIGN][TYPE]number
```

where `SIGN` is an optional minus sign (ie. "-") and `TYPE` is an optional number type, where "x" means hexadecimal and "#" means decimal. If `TYPE` is not present, the type is assumed to be hexadecimal. Some examples are 123, -#456, xAB4, and -xFFFF.

The subtasks are as follows:

1. Read a null terminated from keyboard into buffer. (week 1)
2. Parse the non numeric part of the string to determine its sign and radix. (week 1)
3. Output the 16 bit number to the console in binary. (week 2)
4. Convert the numeric part of hexadecimal strings to a 16 bit twos compliment number. (week 2)

To start, download `lab6b.asm` from the course website and copy the code you wrote for parts 1 and 2 to where it says to insert that code. Do not copy the wrapper code that you did not write.

Task 3: Add code that displays the value in R3 to the console as a binary string. It should convert R3 one bit at a time, to ASCII. The displayed format should be exactly of the form:

```
Answer: 0000000000000000
```

- Your code should be placed in the file where it says "`Put your code for part 3 here.`"
- Your code must start with a label named "PrintResult".
- On entry to your code,
  R3 will contain the value to output to the console.
- At the end of your code, it must unconditionally branch to a label called "ReadString".

Hint: You can check if the leftmost bit of a number is one, by checking if it is negative. To shift all of the bits one position to the left, multiply the number by two.

Demonstrate your working code to the TA before going on to part 4. Use the values #1024, -#1, #0, #511, #32767, -#32768, #21845 for testing, and to demonstrate your program.

Task 4:

Add code that parses a hexadecimal string and converts it to a 16 bit two's compliment number. Assume that all of the numbers will fit in 16 bits.

- Your code should be placed in the file where it says "`Put your code for part 4 here.`"
- Your code must start with a label named "ParseHex".
- On entry to your code,
  R1 will be -1 if the final result should be negated, otherwise it will be 0
  R2 will contain the address of a null terminated string containing the digits to parse
- On exit from your code,
  R3 must contain the converted number, unless there was an invalid string.
- The code should branch to "PrintResult" after it is done, unless it encounters an invalid string, in which case it should branch to "ParseError".

Hint: xABCD $= (((\text{xA} \cdot 16 + \text{xB}) \cdot 16 + \text{xC}) \cdot 16 + \text{xD}$, also $16x = 2 \cdot (2 \cdot (2 \cdot (2 \cdot x)))$

Demonstrate your working code to the TA. Use the values xFFFF, -0, -xFFFF, ABCD, x5555, x3, 30, x1Z11 for testing, and to demonstrate your program.