CS 240
Fall 2019

Final Project
SIC/XE Assembler

Assigned: Monday, November 11, 2019
Due: 10:00 p.m., Tuesday, December 3, 2019 (pass 1 of the assembler)
Due: 12:00 a.m., Saturday, December 14, 2019 (that's 1 second after 11:59:59 p.m. on Saturday)

In this assignment, you will explore the many intricacies of assembling. Your task is to write an assembler for the SIC/XE architecture described in class. Your assembler should be invoked with the following command:

```
sasm program.asm program.obj
```

in order to assemble the program in `program.asm`. Your assembler should place the resulting object code in the file `program.obj`. Your assembler should handle all valid SIC and SIC/XE instructions. It should give priority to addressing modes, by decreasing priority, as follows: extended, based, direct, pc-relative, and finally SIC.

Your assembler should produce object code for all valid SIC/XE programs. If the source code is not valid, your program should print an error message. In addition, it must return an exit code of 1 to the shell (otherwise, it should use exit code 0).

The object code should use a binary version of object records described in class:

- Header: `H`*`NNNNNNMMMLLL`*
- Text: `T`*`AAASBBB...BBB`*
- End: `E`*`EEE`*

Where:

- *`NNNNNN`* denotes a 6-byte, left-justified, space-padded name of the program,
- *`MMM`* denotes a 3-byte binary encoding of a 20-bit starting address,
- *`LLL`* denotes a 3-byte binary encoding of a 20-bit program length,
- ***`AAA`*** denotes a 3-byte binary encoding of a 20-bit address,
- *`S`* denotes a 1-byte binary encoding of the length of data in the text record,
- *`BBB...BBB`* denotes 1-64 bytes, and
- *`EEE`* denotes a 3-byte binary encoding of 20-bit entry point for the program.

Your program should emit "full" text records (as long as possible without padding). **When in doubt of how to assemble an instruction, you may examine the output of my reference assembler.**

How to submit pass 1. You must submit the first pass of the assembler by the pass 1 deadline. This assignment will collect `sasm`.

How to submit the full assembler. This assignment will collect two things: `sasm` (your assembler source code), and `tests` (a directory full of test programs). You may submit as many times as you want up to the deadline. Your final submission will be used in grading.

Submit will not be open until 48 hours before the assignment is due. You may submit as many times as you want up to the deadline. Your final submission will be used in grading.