Term Project - SIC/XE Assembler
Phase (2)

The term project is to implement a (cross) assembler for (a subset of) SIC/XE assembler, written in **C/C++,** producing code for the absolute loader used in the SIC programming assignments.

In phase 2 of the project, you are going to build on the previous phase and use its output to implement pass 2 of the assembler.

Specifications

a) The assembler is to execute by entering

assemble <*source-file-name*>

b) The source file for the main program for this phase is to be named assemble.cpp

c) The output of the assembler should include (at least):

1. Object-code file whose format is the same as the one described in the text book in section 2.1.1 and 2.3.5.

2. A report at the end of pass2. Pass1 and Pass2 errors should be included as part of the assembler report, exhibiting both the offending line of source code and the error.

d) The assembler should support:

1. EQU and ORG statements.

2. Simple expression evaluation. A simple expression includes simple (A <op> B) operand arithmetic, where <op> is one of +,-,*,/ and no spaces surround the operation, eg. A+B.

Bonus

1. General expression evaluation.
2. Literals (Including LTORG)
=C'<ASCII-TEXT>' , =X'HEX-TEXT' , =<DECIMAL-TEXT> forms.
3. Control sections

Notes:

1. Assigned: Tuesday, April 21st.
2. Due date: Sunday, May 10th **8 am**.
3. All members should work together. There is a grade on distributing the load evenly.
4. All member should understand all components in the project, not just the parts they implemented.
5. Cheating will be severely penalized. Both copies will be graded zero. So, delivering a partially functional implementation is much better than delivering a copy.
6. **No** late delivery is accepted.

Deliverables:

- Source Code
- Executable file for pass2 which runs by entering: “assemble <source_file_name>”
- Report that contains:
  - Requirements specifications.
  - Design
  - Main data structures
  - Algorithms description
  - Assumptions (if any)
  - Sample runs.
- You should submit the deliverables in a zipped file with the format: groupNumber_phase2.[rar/zip/...etc]. (for example: "1_phase2.rar") to csed.system.programming@gmail.com