Final Term Project - SIC/XE Assembler
Phase 2

The term project is to implement SIC/XE assemble that produces code for the absolute loader used in the SIC/XE 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

1. The output of the assembler should include (at least):
   a) 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.
   b) A report at the end of pass2. Pass1 and Pass2 errors should be included as part of the assembler report, exhibiting both the erroneous lines of source code and the error.
2. The assembler should support:
   a) EQU and ORG statements.
   b) 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:

- You should work in groups of 4-5 members.
- All team members should work together. There is a grade on distributing the load evenly.
- All members should understand all components in the project, not just the parts they implemented.

- Cheating will be severely penalized. Both copies will be graded zero. So, delivering a partially functional implementation is much better than delivering a copy.

Deliverables:

- Source Code
- Report that contains:
    - Requirements specification.
    - 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")