Compiler Design Homework 2
Due Before Week 3 Class

Modify Your Homework-1 Scanner to Use FSM

- An example scanner is available to you in Blackboard Module “**Week 2**” under “**Hw1-ScannerWithFSM**” as a zip file.
- Using the instructor’s example scanner modify the **getSymbol** routine to scan for the following list of symbols:

      + – * / < <= == >= > = ( ) # “ // /* */
- Add a list of names for these symbols to the end of the enumerated Kind list in **Token.h**
- This modified **getSymbol** scanner must have a **Finite State Machine** (FSM) similar to the one used by the **getKeyword** routine
- This will require making an include file (similar to **ASCII_KW.h**) for the tables for the symbols. Please name this file **ASCII_SYM.h**.
- Submit the files needed for this assignment in a directory called **ScannerWithFSM** with the code for the scanner you have written.

Add Scanner and Preliminary Parser to Spreadsheet

- Write a Spreadsheet Scanner and Preliminary Parser for a small spreadsheet. A skeleton **Spreadsheet** is available to you in the “**HW2-Code**” zip file from Blackboard. You should only need to add code in **Scanner.cpp**.
- The spreadsheet has six columns (of cells) labeled A through F each seven characters wide when printed and separated by a vertical bar, ‘|’, and the spreadsheet has ten rows labeled 0 through 9.

  Note: This spread sheet example will be used for homework assignments for most of the semester.
- The following lines are exemplary of the lines you must parse:

  # Some comment that is totally ignored up to the CR

  A1 “Text”
  A2 -142536
  B3 = B2 + A2;
  A2

  These lines are a **comment** line, a **blank** line, a **text** line, a **numeric** line, an **equation** line and a **clear** line respectively. When scanning these lines ignore leading white space.
- The preliminary white space and the initial ID, the white space following and the look ahead character to identify the line kind may be scanned by hand written code**.** You can also scan the text lines, numeric lines and clear lines in the **scanLine** routine, but the equation lines should use the **parseEquation** function because we will expand on it in future homework assignments.

- For the equation line, the **scanner** must recognize:
    - The symbols: ‘+’ ‘–‘ ‘*’ and eventually ‘/’ ‘(‘ ‘)’ ‘#’
    - digit **=** [**0-9**]
    - **<ID>** = [**A**-**F**](digit)
    - **<NUM>** = (digit)+
- **Note** that unlike the numeric line, the <**NUM**> token is a **non-negative whole number**.
- All **ID**s are a single (case sensitive) uppercase letter (**A**-**F**) followed by a single numeral (**0-9**). Initial IDs are followed by at least one white space character even for the clear line.
- Every **non-blank, non-comment line** in the input file should begin with an **ID** that can be associated with one of the 60 spread sheet cells.
- Text lines will be prefaced with a double quote (**“**). Any terminating quote on a text line can be ignored. The displayed text will be limited to the first seven characters. The text to be displayed should end before the closing quote, end of line or at the seventh character whichever is first**.**
- All numeric lines will be integers possibly prefaced with a minus sign (–) with no white space between the minus sign and the digits.
- Equation lines will be prefaced with an equal sign (=) and (this week) will have **exactly two** IDs separated by **one** operand (+, –, *). (We will make this a little more complete in the following weeks.) Ignore white space within the equation line.
- For HW2 all IDs in an equation must reference cells in previous rows or cells to the left in the same row. (This restriction will be removed next week.)
- Any cell having a scanning or parsing error is an **error cell**. Any equation line referencing a blank line, a text line or an error cell is also an error cell.
- After all the inputs are read, the value of the cells entered should be calculated, and the resulting values should be displayed. Any non-error cell that has not been assigned text or a value (either entered or calculated) should display a blank. Any error cell should display **ERROR**.
- Use the input file **HW2-Input.txt** to generate the output file to be handed in. The other files (e.g. HW2-Test-1.txt) are available for your use if you want to use them to test error conditions.
- If you find any mistakes in the spreadsheet skeleton you were given, please email me with the correction so it can be distributed to the whole class.
- Submit a **zip** file with a **readme** file, your **ScannerWithFSM** directory with all the **code** you have written for the scanner with FSM, your **Spreadsheet** directory with all the **code** you have written for the spreadsheet, your spreadsheet **executable** (identify the OS in the readme), and the **output** file your code generated**. Submit your zip file to Blackboard before the Week 3 class.**

Read Louden Chapter 3 through 3.6.3 (skip 3.5.2) (Parsing) and 4.0 through 4.1.3 and 4.4 (Recursive Descent Parsing)

Submit all results to Blackboard before the Week 3 class.