CISC/CMPE 458 Course Project Winter 2023
Phase 1. Scanner/Screener
K. Jahed – January 2023

In this phase you will undertake the modifications to the Scanner/Screener phase of the PT Pascal compiler to turn it into a Scanner/Screener for Quby. The hardest part of this phase is getting used to using Linux, PT Pascal and S/SL. The actual changes are relatively easy once you figure out how everything works.

Because the PT scanner/screener is run as a co-routine with the parser all in one pass (called *parser.out*), it's necessary to take some special steps to test the result of your modifications. Instead of simply building the parser using the make command, you will build it using the command make scanner, which will make a special version of *parser.out* that runs the scanner/screener only. You will use the –t1 and –o1 flags of the *ptc* command to run your scanner/screener, and the *ssltrace* command to print out the tokens it recognizes – ask your TA for details.

Suggestions for implementing Phase I
The modification of the Scanner/Screener to implement Quby should be relatively straightforward, so no particular implementation hints will be given for this phase. You will need to change the files *scan.ssl*, *parser.pt*, and *stdIdentifiers* in the parser subdirectory. The following is a list of the changes that are likely to be required.

Keywords
Remove the old PT keywords:
not until program const procedure begin repeat
Add the new Quby keywords:
using val def break when module unless elsif

Replace the old PT Pascal predefined identifier:
char
With the new Quby predefined identifier:
String

Character Classes
Add new Quby input character classes (e.g., *lPercent*) for the characters percent ("%"), hash (“#”), exclamation ("!"), question {"?") and dollar ("$") used in Quby. Reuse the existing *lQuote* character class to refer to the Quby double quote (") instead of the PT Pascal single quote ('). Don't forget to update the initialization of the character class map in *parser.pt*.

Syntax Tokens
Add new Quby syntax tokens (e.g., *pElsif*) for the Quby keywords **using**, **val**, **def**, **unless**, **elsif**, **break**, **when** and **module**, and for the new Quby syntax symbols double equals ("=="), not ("!"), question ("?"), dollar ("$") and hash ("#"). Re-use the old *pNotEqual* syntax token for the new Quby not equals symbol (“!=”).
Remove the old PT syntax tokens for the keywords **not**, **until**, **program**, **const**, **procedure**, **begin** and **repeat**, and the PT colon equals (":=").

String Literals
Replace the scanning and screening of PT single-quoted char literals (e.g. 'hi there') with the scanning and screening of Quby-style double quoted string literals (e.g. "hi there"). Reuse the old PT character class *lQuote* to represent the double quote character and reuse the old PT char literal compound token *pLiteral* to represent the new Quby double-quoted strings. Remember that in Quby, the null string ("") is allowed.

Comments
Replace the scanning of PT { } and (* *) comments with the scanning of Quby % to end of line comments. Be careful to preserve the right number of *pNewLine* tokens in the scanned output when skipping % comments.

What to Hand In
Submission of your assignment will be electronic - ask your TA for details. Hand in the following:

(1) Documentation of the modifications you have made to the scanner/screener, including a list of the changes made to the S/SL source, a

description of the changes to the token streams and tables, and an explanation of any new error signals. The documentation should be easily understandable to someone who already knows the basics of the PT Pascal compiler. It should be complete but not verbose.

(2) A complete copy of your Quby compiler so far with the changed scanner/screener source in the parser subdirectory, with changes indicated by appropriate comments. Your compiler should be ready for us to compile and run using the "make scanner" command to compile
your source so far.

(3) A suite of test inputs (small Quby programs) designed to demonstrate the (partial) correctness of your scanner/screener by forcing it through every new or modified logic path in the S/SL source at least once. The test suite should be accompanied by comments indicating the purpose of each test input. Modifying the old PT test suite in the test/ subdirectory of the PT Pascal source to be Quby programs may help as a start.

The purpose of what you hand in is to clearly indicate exactly what you have done, provide enough information about the changes to allow further maintenance of the new scanner/screener by someone who already knows the PT Pascal compiler, and to convince them that the changes you have made are complete and correct according to the specification of Quby.

Details of exact expectations and electronic submission will be given in tutorial. See the handout "Project Requirements" for marking criteria and general expectations on project results.