14th Sep 2016

Bison – Clite

Compilers 2016 Assignment 2 | **Deadline: Tuesday, 20-09-2016, 11:59 p.m.**

For the second assignment (as well as phase 1 of your project), you have to build a parser. For this assignment, you'll make use of the **bison**, an open source tool, and build a parser for **C-lite**.

Important links:

- Documentation for bison
- C-lite Grammar

The objective of this assignment is to implement a parser for a subset of the C-lite grammar as given below.

[], {} -> optional, grouping ("bold")

[], {} -> part of the syntax

'|' -> "OR" ie option amongst rules

Program	->	int main ( ) { Declarations Statements }
Declarations	->	**{** Declaration **}**
Declaration	->	Type Identifier **[** [Integer] **]** ;
Type	->	int | bool
Identifier	->	Letter **{** Letter | Digit **}**
Letter	->	a | b | … | z | A | B | … | Z
Digit	->	0 | 1 | … | 9
Integer	->	Digit **{** Digit **}**
Statements	->	**{** Statement **}**
Statement	->	; | Block | Assignment
Block	->	**{** Statements **}**
Assignment	->	Identifier **[** [Expression] **]** = Expression ;
Expression	->	Term **{** AddOp Term **}**
Term	->	Factor **{** MulOp Factor **}**
AddOp	->	+ | -
Factor	->	Identifier **[** [Expression] **]** | Literal | (Expression)
Literal	->	Integer | Boolean
MulOp	->	/ | * | %
Boolean	->	true | false

Your program should successfully parse all valid programs, and throw an error if it encounters an invalid program. Ideally, your program should also ignore comments.

Output to stdout:

"Success" -> Successful parse

"Syntax error" -> Invalid program

Create a file bison_output.txt which on encountering a rule, prints the respective output on a new line. Only the following statements need to be processed.

Output	Rule Found
Program encountered	Program [output 1 line]
Int declaration encountered Id = <Identifier>	Int Identifier; [output 2 lines]

Int declaration encountered
Id=<Identifier>
Size=<Integer>
	Int Identifier [Integer]; [output 3 lines]

Boolean declaration encountered Id = <Identifier>	Bool Identifier; [output 2 lines]

Boolean declaration encountered
Id=<Identifier>
Size=<Integer>
	Bool Identifier [Integer]; [output 3 lines]

Assignment operation encountered	Identifier **[** [Expression] **]** = Expression ; [output 1 line]
Addition expression encountered	Expression + Expression [output 1 line]
Subtraction expression encountered	Expression - Expression [output 1 line]
Division expression encountered	Expression / Expression [output 1 line]
Multiplication expression encountered	Expression * Expression [output 1 line]
Modulus expression encountered	Expression % Expression [output 1 line]
Integer literal encountered Value=<value>	Integer literal [output 2 lines]
Boolean literal encountered Value=<value>	Boolean literal [output 2 lines]

Please stick to the output format as there will be no manual evaluation of codes submitted.

Submission Format:

Compress a) the flex code (named Assignment2.l), b) the bison code (named Assignment2.y), c) a sample input test case (named test_input), and d) a readme file and upload the zip file. The output file generated must be as specified above.

The zip file should be named rollno_Assignment2.zip.