CS 4850 Flex Bison Assignment

Submit files to e-learning due Oct. 22 before class.

Warning! If you copy any of your grammar from another source you must include exact reference as to 1. Where you got it and 2. Why you think it is correct.

Implement expressions Using Flex grammar and Bison grammar with actions.

<expression> is 1 + 2 or 1 < 2 or “hello” > “goodby” or 1.2 / 3 or 12 or “hello” + “ goodby”, or (1 + 5) / 3; the operators including Boolean for all 3 data types.

Also the Boolean operators And, Or, XOR; Examples: 1 < 3 aNd !(3 < 4 XOR 4 < 5). Remember not cAsE sensitive.

+, -, \*, /, %, <, <=, >, >=, ==, !=

Integer, real, string. String + String and String {Boolean operators} String

No variables, no assignment, if-elseif-endif. Only Honk the results of executing an expression to stdout. Read from stdin, write to stdout.

Read a line, evaluate and honk the result, read a line … until end of file.

> 1 / 2.0

0.5

…

Plan:

Work on this before class Tuesday so you can bring questions.

Write a grammar input file for Bison for expressions.

Decide how to pass the values of different data types from Flex to Bison and around in Bison!

Remember you will be adding assignments to variables soon:

Adding Symbols for statements like

A = 1

A = A + 1

honk A

2

Test your code

Submit your Flex and Bison grammar files (and other source files)

Do not #include c code in other .c files just #include .h files, compile and link your files, do not include c files in each other.

**Along with a Makefile** which will run flex and bison to create the Bison and Flex .c files and compile the whole project on the CS department’s Linux servers or similar (gnu) gcc based C compiler. To run the code from the makefile put a run: target so you or the grader can type “make run”. The main target should just link the .o files into an executable file.

Also include a readme file with any instructions you think the grader could use.

Make sure you do not link to any non-standard libraries on the command line. In other words standard C90 or C99 . You can putty/ssh into the cs department’s servers or take code to C-208 to test it.