CS 4850 Second Bison Assignment

Submit a zip file to e-learning due Friday 10/24 11:30 PM. Grammar files, Makefile and your .c source code file(s). So not submit files generated by Flex or Bison.

Add to the grammar from the first part below.

Add symbols for variables and the assignment statement. ‘=’

A = 23 ?

B = A % 20 ?

HORN B ?

A = “hello “ + “world” ?

HORN A ?

I suggest your put the symbol handling in another file and use functions to access the table so you can change the way it works when we add functions and other symbols later.

Use the same header file and defined tokens for your symbol table you are using for Bison and Flex.

## First Part Below

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 Horn <expression> ? Using your previous Flex grammar and new Bison grammar with actions.

<expression> is 1 + 2 or 1 < 2 or “hello” > “goodby” or 1.2 / 3 or 12

No variables, no assignment, no functions yet. Only horn the results of executing an expression to stdout.

HORN

1

/

2

?

O

HORN 1 / 2.0 ?

0.5

…

Plan:

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

Transfer your tokens used in the Flex assignment to Bison #token ….

Also add tokens to Bison and add to flex grammar for key word(s) added to the language like ‘horn’ and ‘?’ remember language is NOT CASE SENSITIVE.

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

Test your code

submit your Flex and Bison grammar files (and any other source files)

**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. Make sure you do not link to any libraries and use the –Wpedantic and –ansi switches.

In other words standard C as is.