BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CSE310 (Compiler Sessional), January 2019 Term All Lab Sections, July 1, 2019

1 Introduction

In the previous assignment, we have constructed a lexical analyzer to generate token stream. In this assignment we will construct the last part of the front end of a compiler for a subset of C language. That means we will perform syntax analysis and semantic analysis with a grammar rule containing function implementation in this assignment. To do so, we will build a parser with the help of Lex (Flex) and YACC (Bison).

2 Language

Our chosen subset of C language has following characteristics.

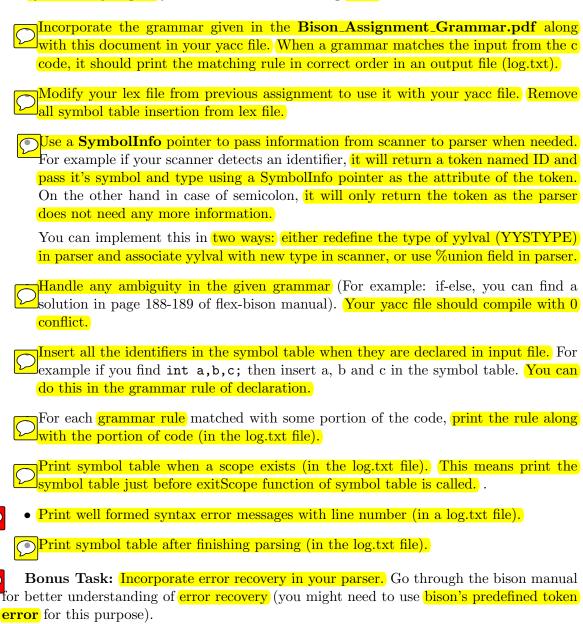
- There can be multiple functions. No two function will have the same name. A function need to be defined or declared before it is called. Also a function and a global variable cannot have the same symbol.
- There will be no pre-processing directives like include or define.
- Variables can be declared at suitable places inside a function. Variables can also be declared in global scope.
- All the operators used in previous assignment are included. Precedence and associativity rules are as per standard. Although we will ignore consecutive logical operators or consecutive relational operators like 'a && b && c' 'a < b < c'.
- No break statement and switch-case.

3 Tasks

You have to complete the following tasks in this assignment.

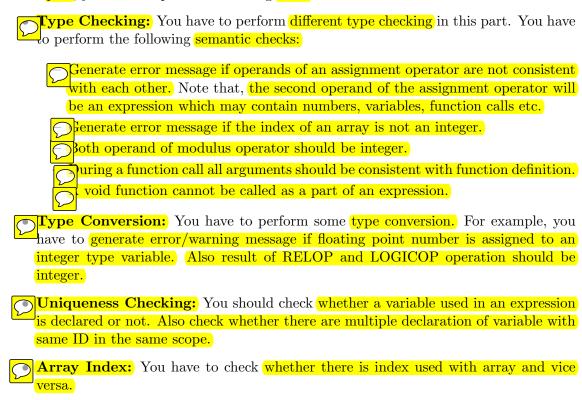
3.1 Syntax Analysis

For syntax analysis part you have to do the following tasks.



3.2 Semantic Analysis

In this part, you have to perform following tasks:

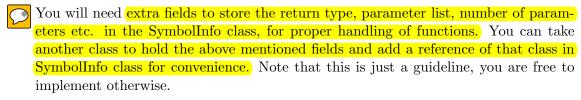


Function Parameter: Check whether a function is called with appropriate number of parameters with appropriate types. Function definitions should also be consistent with declaration if there is any. Besides that, function call cannot be made with non-function type identifier.

To implement this task, you can add necessary fields in the SymbolInfo class as required but try to avoid redundant fields.

3.3 Handling Grammar Rules for Functions

For implementing the grammar rules of functions, you will need to add some fields in your SymbolInfo class.



As a part of semantic analysis you have to match the function declaration and function definition and report an error if there is any mismatch in return type, parameter number, parameter sequence or parameter type as well as invalid scoping of the function.

4 Input

The input will be a C source program in .c extension. File name will be given from command line.

5 Output

In this assignment, there will be two output file. One file, log.txt, will contain matching grammar rules, corresponding segment of source code and symboltable entries as instructed in previous sections. Another file, error.txt will contain error messages with line number. For any detected error print something like "Line no 5: Corresponding error message". Print the line count and no of errors at the end of log file.

For more clarification about input output check the Sample I/O files.

6 Submission

- Plagiarism is strongly prohibited. In case of plagiarism -100% marks will be given.
- No submission after the deadline will be allowed.
- Deadline will not be extended in any situation.

7 Submission

All Submission will be taken via moodle. Please follow the steps given below to submit you assignment. 10% out of the total assignment marks will be allocated for the correct submission.

- 1. In your local machine create a new folder which name is your 7 digit student id. Do not miss this point.
- 2. Put the lex file named as <your_student_id>.l and yacc file named as <your_student_id>.y containing your code. Also put additional c file or header file that is necessary to compile your lex file. Do not put any input/output file, any shell script, the generated lex.yy.c file, or any executable file in this folder.
- 3. Compress the folder in a zip file which should be named as your 7 digit student id.
- 4. Submit the zip file within the Deadline.

8 Deadline

Submission deadline is set at July 20 (Saturday), 10 pm. there will be no extension of this deadline. In case you can not access the moodle, send an e-mail to any of your course teachers attaching the zip file mentioned in earlier section (within the deadline of course).