CSL3040: Programming Assignment 2

Submission deadline: Nov 05, 2022

Instructions:

1. Plagiarism is prohibited. It would lead to zero marks for this assignment.
2. This is a group assignment. You have to form groups of two and any one member has to submit. Clearly write the name of the group members in the README file.
3. You should submit a single ZIP file with the following naming convention *B20CSxxx_B20CSxxx_PA2.zip*

This PA is designed to get you acquainted with Lex and Yacc. You can refer to the attached resources or use Online resources to get an understanding of the syntax. The description of the assignment is as follows:

1. Write a LEX program to generate tokens for C language.
   - The input program may contain comments.
2. Define a grammar for C language and write a YACC program to check whether your grammar can successfully parse any given C program.

   Define rules for:
   - Conditions: All if-else statements, switch-case (optional).
   - Loops: for and while, do-while is optional.
   - Arithmetic and logical statements, declaration, print, scan and return statement.
3. Write a program in YACC to generate the symbol table for the C program, if the program can be successfully parsed.
4. Write a program in YACC to generate the syntax tree for any C program.

A sample input file "**input.c**" is attached with the assignment. You could use it to test your code wherever required. Note that "input.c" is just a sample file. Your code should work for any input program. You can consider an input program to consist of only one explicit function, i.e., main, and have assignments, conditions, loops, and expressions. Submit the source code for all the parts of the assignment separately.

Readme file:

Also, submit a Readme file containing instructions to run your programs. If you make any assumptions, explain that in detail. Remember that your Readme is the key to us understanding your programs, but it should not contain junk. A good Readme is not more than one page long. Your README file should consist of execution steps for each program and the output generated (output for a given input.c is necessary).

Sample Output:

INPUT file

```
#include<stdio.h>
#include<string.h>
int main() {
   int a=0;
```

```
    int y=4;
    for(int k=0; k<y; k++) {
        a = a+4;
    }
    return a;
}
```

OUTPUT for Question 1

```
1       #include<stdio.h>
2       #include<string.h>
3       int
3       main
3       (
3       )
3       {
4       int
4       a
4       =
4       0
4       ;
5       int
5       y
5       =
5       4
5       ;
6       for
6       (
6       int
6       k
6       =
6       0
6       ;
6       k
6       <
6       y
6       ;
6       k
6       ++
6       )
6       {
7       a
7       =
7       a
7       +
7       4
7       ;
8       }
9       return
9       a
9       ;
10      }
```

OUTPUT for Question 2

Parsing Successful. (or Parsing failed)

OUTPUT for Question 3

```
LEXEME               DATATYPE    SYMBOL TYPE
____________________________________________
#include<stdio.h>    N/A         HEADER
#include<string.h>   N/A         HEADER
main                 N/A         KEYWORD
a                    int         IDENTIFIER
0                    int         CONSTANT
y                    int         IDENTIFIER
4                    int         CONSTANT
for                  N/A         KEYWORD
k                    int         IDENTIFIER
return               N/A         KEYWORD
```

Note:- Symbols and operators are not included in the symbol table. For KEYWORD and HEADER, the datatype is N/A.

OUTPUT for Question 4 (you can use one of the format)

Preorder Tree traversal (Inorder or Postorder is also allowed):
PROGRAM HEADER BODY MAIN DECLARATION DECLARATION CONDITION STATEMENT RETURN

OR

Parse Tree

```
                              PROGRAM
    HEADER                       MAIN
             DECLARATION   DECLARATION   CONDITION   RETURN
                                          STATEMENT
```