CSCI 4200 Fall 2024

Project 1

100 points

**1 Note**

This project is to ask you to understand an existing lexical analyzer, and then re-write it by Java.

**2 Description**

Modify the front.c lexical analyzer program, given in Section 4.2 of the textbook (also attached in the assignment. The program is based on the following grammar.

<assign> → IDENT = <expr>

<expr> → <term> {(+ | -) <term>}

<term> → <factor> {(\* | /) <factor>}

<factor> → IDENT | INT\_LIT | ( <expr> )

Recall that the nonterminal symbols are enclosed in angle brackets <>.

a. Fully understand the front.c lexical analyzer program as is given on Pages 166 – 170 of your textbook and in the attached PDF file.

b. Modify the program to display the tokens in a textual form rather than numeric form. For example, instead of displaying

Next token is: 25 Next lexeme is (

display

Next token is: LEFT\_PAREN Next lexeme is (

**3 Tokens**

Use the following token names instead of numbers in your implementation:

ADD\_OP

ASSIGN\_OP

DIV\_OP

END\_KEYWORD

END\_OF\_INPUT

IDENT

INT\_LIT

LEFT\_PAREN

MULT\_OP

PROGRAM\_KEYWORD

RIGHT\_PAREN

SEMICOLON

SUB\_OP

**4 Example Input and Output**

For example, if the input is:

(sum + 47 ) / total;

Then the output is:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Next token is: LEFT\_PAREN Next lexeme is (

Next token is: IDENT Next lexeme is sum

Next token is: ADD\_OP Next lexeme is +

Next token is: INT\_LIT Next lexeme is 47

Next token is: RIGHT\_PAREN Next lexeme is )

Next token is: DIV\_OP Next lexeme is /

Next token is: IDENT Next lexeme is total

Next token is: SEMICOLON Next lexeme is ;

Next token is: END\_OF\_INPUT Next lexeme is EOI

Lexical analysis of the program is complete!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

For another example, if the input is:

Area = PI \* Radius \* Radius;

Then the output is:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Next token is: IDENT Next lexeme is Area

Next token is: ASSIGN\_OP Next lexeme is =

Next token is: IDENT Next lexeme is PI

Next token is: MULT\_OP Next lexeme is \*

Next token is: IDENT Next lexeme is Radius

Next token is: MULT\_OP Next lexeme is \*

Next token is: IDENT Next lexeme is Radius

Next token is: SEMICOLON Next lexeme is ;

Next token is: END\_OF\_INPUT Next lexeme is EOI

Lexical analysis of the program is complete!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Once you finish the program, please write a lab report of word document after running your source code. The lab report should have the following two parts:

1. The screenshot of running result. The screenshot should be clear enough to recognize your running result. You don’t need to show both examples as in this document in your lab report. One is enough (either (sum + 47 ) / total; or Area = PI \* Radius \* Radius; ).
2. Analysis. If your program has limitations, please specify the limitations. Otherwise say the program meets the requirement.

Note: If your computer has no Microsoft office, please use virtual lab to connect a UNG computer to use Word in Microsoft Office or submit a PDF file. Other formats are not accepted.

Note:

You can use spaces between tokens. For example, (  sum  +  47  )   /  total  ; is fine.

Area  =  PI  \*  Radius  \*  Radius ; is fine.

You don't need to use file operations. You just need to read an input string and display the output of the tokens.

The output must be the same as my two given examples, including  the spaces, the lines of \*, etc.

One is: (sum + 47 ) / total;

The second one is: Area = PI \* Radius \* Radius;

Please see the detailed manner of output.

**5. Grading Rubrics**

|  |  |  |
| --- | --- | --- |
| **Category** | **Description** | **Points** |
| Program Readability | Sufficient and readable comments (don’t overdo it!)  Readable program – descriptive identifier names; use of constant identifiers; appropriate program structures such as loops, decisions, etc; appropriate indentations and spacing, etc. | 25 |
| Program logic | Straightforward (simple) and understandable logic | 25 |
| Output | Correct and complete results. The output must be the same as my given examples, including the spaces, the lines of \*, etc. | 25 |
| Screenshot of running result in Lab report | The screenshot is clear to recognize the running result. | 15 |
| Analysis in lab report | The lab report has analysis if the program has limitations, or the lab report states the program meets the requirement. | 10 |

**Note: your program must be executable; otherwise you will get no more than 40% credit.**

In summary, please submit the following 2 files:

1. A java source code (\*.java file) for the question.
2. A word document of the lab report.