CSE4120 Fundamentals of Compiler Construction (Fall 2022)
Homework 1: Lexical Analyzer (Scanner)

Handed out: Oct 01, Due: Oct 26, 11:59PM (KST)

1. Problem Description

In this programming assignment, you will build a lexical analysis phase for the C- language, defined in Appendix A of the textbook.

2. Your task and requirements (Read Carefully!)

(1) You should first read carefully Chap.2.6 (pages 81-91) and Appendix A.1 (pages 491-492) for this programming assignment. Next, your task is to modify `main.c`, `global.h`, `util.h`, `util.c` (shown in pages 502-511) and "`tiny.l`" (shown in pages 537-538) to build your lexical analyzer for C- language. Note that all files for TINY scanner will be given.

(2) In addition to the program, you should also write a **Makefile**. The TA will build your code by running “make”. It should create the binary files, **`hw1_binary`**.

(3) Your program should read the source code and output the result to files. The input file is **`hw1.c`** which contains the code written in C- language. Your output file should be named **`hw1_20200001.txt`**. The red numbers should be changed to **your student ID**.

(4) The format of the input and output files is as follows:

Input example:

```
/* binary_search program */
int arr[11111];
int binarySearch(int x)
...
```

Output example (Both tabs or spaces can be used as white space)**:**

```
line number      token      lexeme
----------------------------------
    2            INT        int
    2            ID         arr
    2            [          [
    ...          ...        ...
```

NOTE: two example codes and corresponding results will be given.

(4) You have to do this assignment on the Unix or Linux system. Also, you should use C and Flex (or Lex) to implement the lexical analyzer.

NOTE: If you do not have Unix or Linux system, I encourage you to use **Google Colab** instead. You can easily install Flex in your Colab project using the following commands:

```
%%shell
apt-get install flex
```

3. Submission

You should submit your codes and the Makefile. You do not need to submit compiled binary files. Combine your files into a zip file named cse4120_hw1_**20200001**.zip. The red numbers should be changed to **your student ID**. Submit your files on the cyber campus.

4. Evaluation Criteria

(1) Your scanner should be able to recognize all tokens successfully.

(2) Your scanner should be able to handle comments `(/* ... */)` and white space.

(3) Your scanner should be able to handle lexical errors.

(4) Your scanner should capture the line number of each lexeme correctly.

5. Notes

- You must write your own code. You may discuss ideas with other students but must not copy their work. Similarly, you can find ideas on the Internet, but must not copy the code from the sources obtained from the Internet.

- Our department uses a software that detects duplicate codes. Since the software uses an assembly code level detection, just changing the variable names will not bypass the software. Make sure you write your own code from the given file set. Then, you will have no problem.

- Your program should be able to compile and run on the Linux system. Make sure you check before you submit your work. The TA will download your code, run “make” and execute your binary files to check if they produce correct outputs.

6. Late Policy

No late turn-ins will be accepted for this assignment.