Project #1. Scanner Report

2017030328. 조지훈

**Compilation method and environment  
사용 에디터: Visual Studio 2017 & VIM editor**

**개발 OS: Ubuntu 18.04.1 LTS (Windows 10 PRO 21H1버전에서 WSL2 이용)  
컴파일러: gcc 7.5.0  
장치 사양  
프로세서 Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz   
RAM 16.0GB  
시스템 종류 64비트 운영 체제, x64 기반 프로세서**

**Explanation about how to implement and how to operate**

**Tiny compiler source code를 기반으로 수정하여 Scanner 구현  
6개의 Reserved Words(Keywords)와  
19개의 Symbol, 그리고 ID와 NUM을 token으로 인식할 수 있도록 구현  
/\* \*/ 해당 Symbol은 Comment로 처리하도록 되어 있음**

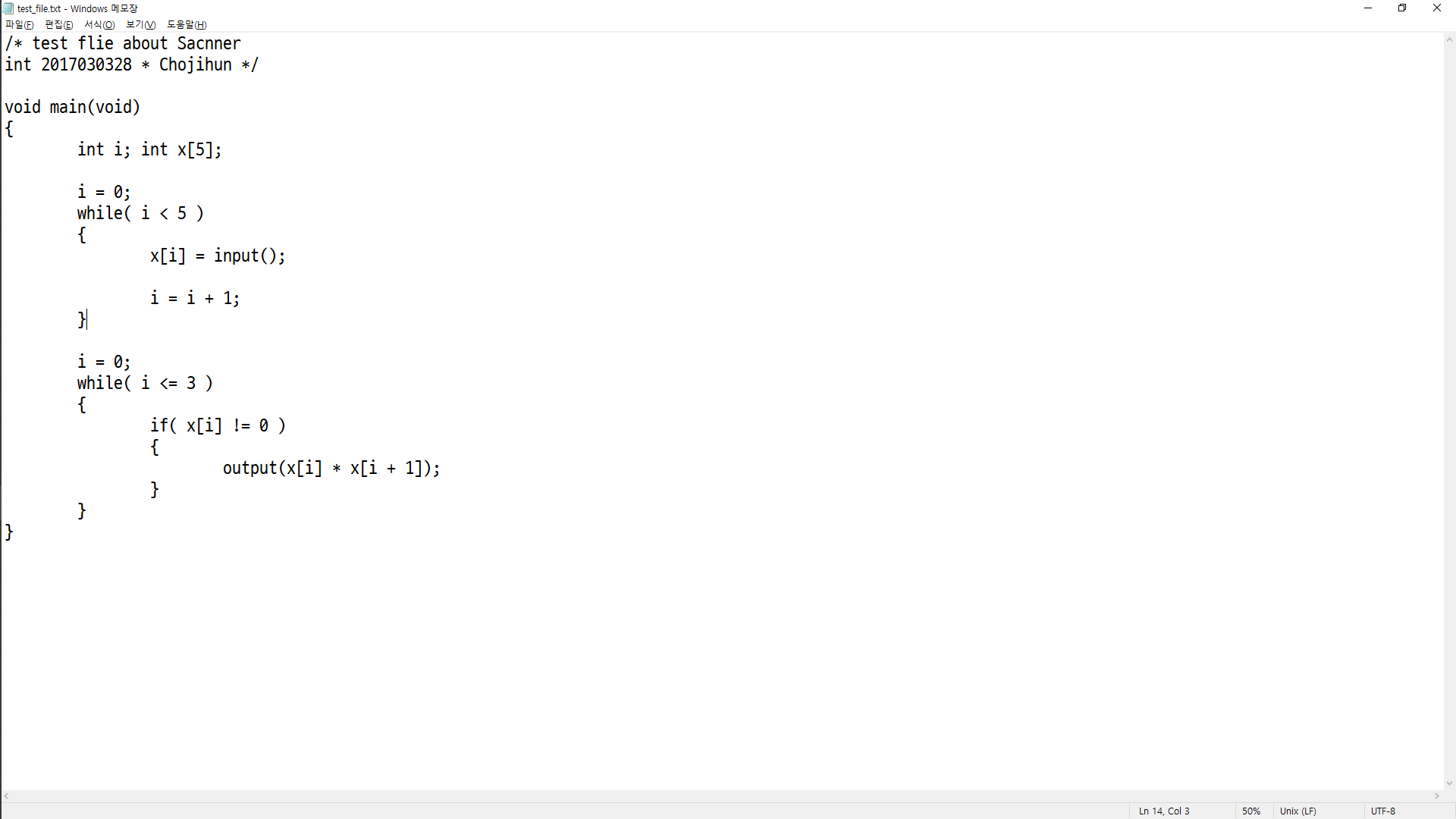
**./cminus\_cimpl [Scan할 file 이름]  
./cminus\_lex [Scan할 file 이름]  
을 해서 Scanner를 동작시킬 수 있음.**

**기존 Tiny compiler의 main.c에서 NO\_PARSE, TraceScan을 TRUE로 설정하고, globals.h에서 Token-Type을 C-Minus에 맞게 새로 설정, utils.c에서 globals.h에서 설정한 TokenType과 연관되어 작동될 수 있도록 printToken()을 수정하였음.**

**DFA를 이용한 방식에서는 scan.c를 수정하여 getToken()에서 읽는 값에 따라 C-Minus Token에 맞게 StateType state를 바꿔주도록 수정하였고, 그에 따른 결과인 TokenType currentToken 역시 따라 C-Minus에 맞게 수정하는 방법으로 구현**

**Regular Expression을 이용한 방식에서는 Lex(flex)를 이용. flex를 설치하고, Tiny compiler의 tiny.l를 이용하여 cminus.l를 작성. Rule Section을 C-Minus에 맞게 수정하는 방법으로 구현**

**Example and Result Screenshot  
- Example**



**- Method 1  
Result of test\_file.txt**

**- Method 2  
Result of test\_file.txt**