Project 3
Static Type Checker

Type checking is the processes of identifying errors in a program based on explicitly or implicitly stated type information. In the third project, you need to implement a type checker for your C subset. The checker needs to:

- Create a symbol table.
- Insert the type of each variable into the symbol table.
- Perform the type checking.

The basic type checking rules are summarized as follows. You can add your rules for your C subset.

(1) Each variable must be declared before it is used.
(2) Each identifier can be only declared once.
(3) The types of the operands of an operator must be the same.
(4) The types of the two sides of an assignment must be the same.
(5) ==、!=、>=、>、<=、<等運算的結果，其 type 為 boolean.
例如: expression 2>3 的 type 為 boolean.
(6) if-else、for-loop、while-loop constructs 的 condition 部分，其 type 必須是 boolean，否則為 type error.
(7) 其他規則可自行增加 (your rules for the C subset)

The type checker needs to report an error message for each type error detected. Each type error message should contain **the line number** where the error is detected and **an explanation of the error**. The format for printing a type error message is as follows:

Type Error: line number: the error message.

For a sample C program given as follows:

```
1. void main()
2. {
3.    int num;
4.    int s;
5.    int index;
6.    float s;
7.
8.    k = 0;
9.    num = index + 3.21;
   }
```

The type checker will report the following error messages:

Type Error: 6: Redeclared identifier.
Type Error: 8: Undeclared identifier.
Type Error: 9: Type mismatch for the two sides of an assignment.

In your hand-in report, you need to have the followings:

- **Describe your type checking rules for your C subset.**
- Give a set of testing programs which can illustrate the features of your type checker. (at least 3 test programs)
- Use the "**ANTLR**" to help you develop the parser.
- You can use **Java** or **C** to write your parser. (Java is recommended)

Please turn in the following:

- A file describes your type checking rules and your C subset. (MS-WORD file)
- The source codes:
  - ANTLR grammar file, `myChecker.g`.
  - A program to call your static checker, `myChecker_test.java`. (or `myChecker_test.c`.
  - Testing programs. (at least 3 programs)
- A readme file (pure text file) describes how to compile and execute your type checker.
- A "Makefile".

Due Date: May 30 (Saturday), 24:00pm, 2020.