Compiler-2018 Project2

1. Changes to my previous scanner

- Add return tokens so that Yacc can use them.
- Add `#include "y.tab.h"` to the front of *lex.l*

2. Ability of parser

The parser will check syntax correctness

- Tokens passed to the parser
  - Delimiters

    `,` `;` `()` `[]` `{}`
  - Arithmetic, Relational, and Logical Operators

    `+` `-` `*` `/` `%` `=` `<` `<=` `!=` `>=` `>` `==` `&&` `||` `!`
  - Keywords

    `while` `do` `if` `else` `true` `false` `for` `print` `const` `read` `void` `continue` `break` `return`
  - Data type keywords

    `int` `bool` `float` `double` `string`
  - Identifiers

    `string of letters and digits beginning with a letter`
  - Integer, Float, Scientific
  - Strings
- Tokens discarded

    - white space, tabs, newlines
    - comments

      `// c++style`

      `/* c-style */`

Create syntax declaration grammer

- Program Units:
    - Program
    - Function declaration and definition
    - Procedure declaration and definition
- Data types and declaration
    - variable declaration
    - const declaration
- Statement
    - compound
    - simple
    - conditional
    - while
    - for
    - jump
    - procedure call (function invocation)

Output syntax correctness

If no syntax error:

```
|-------------------------------|
| There is no syntactic error! |
|------------------------------|
```

When an error occurs, print error message:

```
|-------------------------------------------------------------------------
| Error found in Line #[line number where the error occurs]: [source code of that line]
|
| Unmatched token: [the token that is not recognized]
|-------------------------------------------------------------------------
```

3. Platform to run scanner

Use **lex** and **Yacc** to implement scanner

build and execute in Linux/Unix system

Take Ubuntu as example

- Install Flex/Lex and Bison/Yacc

```
% sudo apt-get install Bison flex
```

3. How to run my code?

- To run my scanner, type

```
% make
% ./parser [inputfile]
```

- To delete files except `lex.l` `yacctemplate.y` `Makefile`, type

```
% make clean
```