Compiler Homework1

Input file: main.c

Output file: token.txt

Compile: **g++ -Wall -W -pedantic -o lexer lexer.cpp**

Execute: **./lexer**

---

*source code可為無空白版本

測資text.c

```
int b;
int main ( ){
    int _a123;
    a=3;
    b=1;
    char c='\n';
    char d='s';
    double e=123.4;
    int f[10];
    if (b<a) { // 12345
        a=a+1;
    }
    else{
        b=0 ;
    }
    b=b*10;
    int 1s2e ;
    return 0 ;
}
```

結果token.txt

```
Line 1:
        <Keyword>       :       int
        <Identifier>    :       b
        <Special>       :       ;
Line 2:
        <Keyword>       :       int
        <Keyword>       :       main
        <Special>       :       (
        <Special>       :       )
        <Special>       :       {
Line 3:
        <Keyword>       :       int
        <Identifier>    :       _a123
        <Special>       :       ;
Line 4:
        <Identifier>    :       a
        <Operator>      :       =
        <Number>        :       3
        <Special>       :       ;
Line 5:
        <Identifier>    :       b
        <Operator>      :       =
        <Number>        :       1
        <Special>       :       ;
Line 6:
        <Keyword>       :       char
        <Identifier>    :       c
        <Operator>      :       =
        <Char>          :       '\n'
        <Special>       :       ;
```

• Variables

char nextChar

- 每一次正在處理的character

int charClass

- character的類別, 用作判別, 有letter, digit, underline, blank, newline, tab和others

char peekChar

- 用peek()先知道nextChar的下一個character是什麼

int peekClass

- peekChar的類別, 有時用來當作終止條件

int lineNum

- 紀錄目前行數的變數

string lexeme

- 將每一次的character都加到lexeme直到確定是某種token再將此lexeme pop

string keywordTable[]

vector<string> keywords

- 作業規定的keyword list

string op1("=!+-*/<>&|")

- 作業規定的operators中可能會出現的第一個字元

string op2("=>&|")

- 作業規定的operators中可能會出現的第二個字元

string special("{}();,[]")

- 作業規定的special list

- 這些string用來和目前的lexeme查表比對有沒有符合的keyword, operator 或special

• Functions

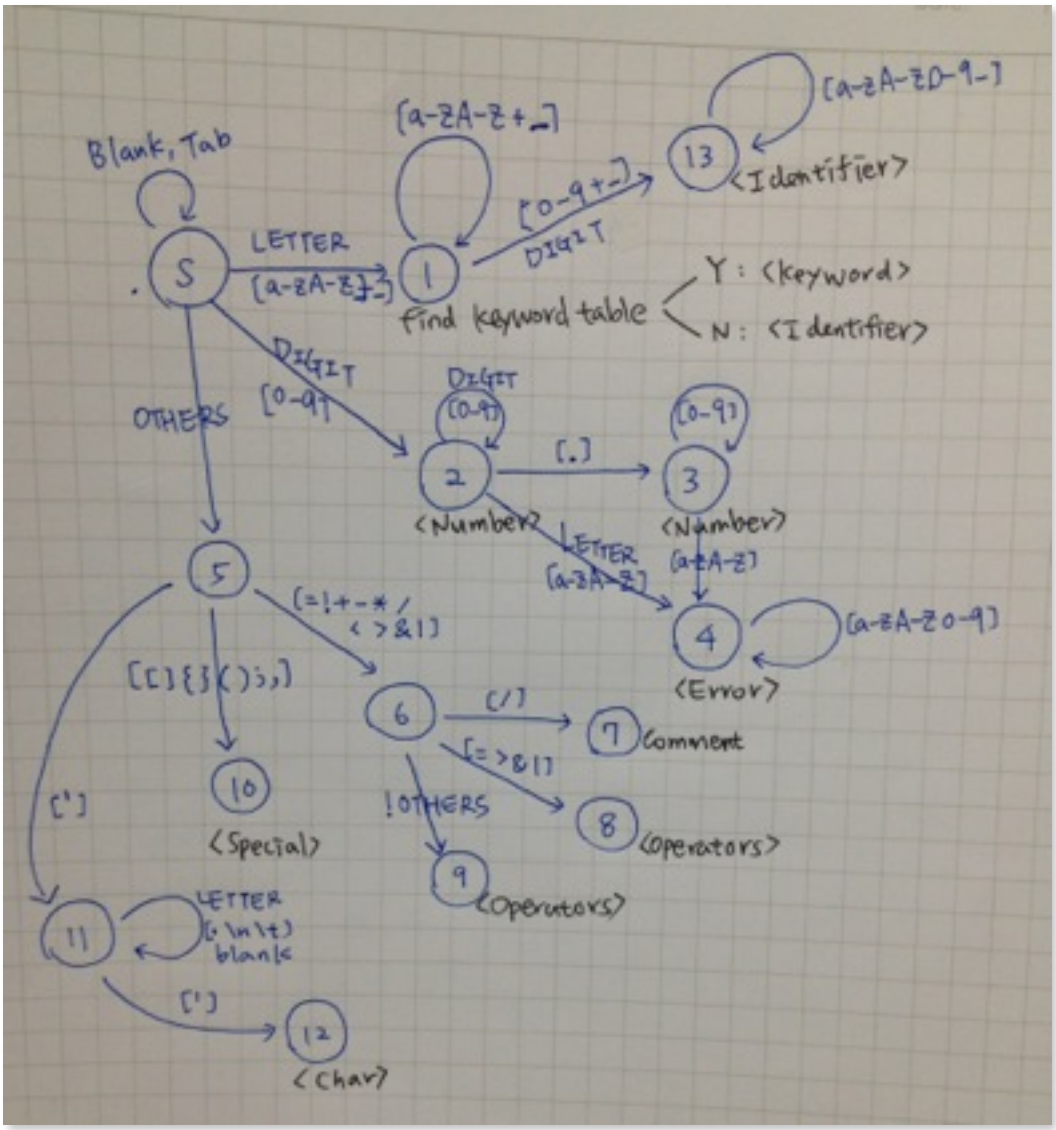
void getChar();

- a function to get the next character of input and determine its character class

void getPeek();

- a function to peek the next of the next character of input but does not use up as well as determine its character class

• 簡略State Diagram



• 程式說明

在主要判別部分, 程式用getChar()一次讀取一個字元, 使用while loop直到檔案末, while loop中的switch依照字元的類別分成7種case.

case BLANK 和 TAB: 略過再讀取下一個字元

case NEWLINE: 遇到換行鍵就在token.txt印出新行數

case UNDERLINE: '_'開頭只可能為identifier, 把字元一一加進lexeme中直到下一個字元不是letter/digit/underline

case LETTER: letter開頭可能為identifier或keyword, 把字元一一加進lexeme

中直到下一個字元不是letter/digit/underline, 再查keywordTable有沒有符合的, 若有相同的即為keyword, 沒有則是identifier

case DIGIT: digit開頭可能為number或數字開頭的命名錯誤, 把字元一一加進lexeme中直到下一個字元不是digit或point, 若下一個字元出現letter, 則是error

case OTHERS: 又分為3種, 若是string op1中的字元, 可能為//註解或1個及2個字元的operator, 用string.find()比對, 若return非npos則比對成功；若為string special中的字元較為簡單, 比對成功則印出；若為單引號, 則是char, 把字元加進lexeme中直到遇到第二個單引號再pop