「編譯器製作」作業一 Readme Simple Java — Scanner

學號：B093040003 姓名：鄭璟翰

一、Lex 版本：flex 2.6.4

作業系統平台：Ubuntu 22.04.2 LTS

```
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ lex --version
flex 2.6.4
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ cat /etc/issue | cut -c1-18
Ubuntu 22.04.2 LTS
```

圖片 1、Lex 版本及作業系統平台

二、執行方式：(1)、make；(2)、./B093040003 < 測試檔案.java

```
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ make
flex B093040003.l
gcc lex.yy.c -o B093040003 -lfl
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ ./B093040003 < Test1.java
```

圖片 2、執行方式

三、如何處理這份規格書上的問題

(一)、Symbols

屬於 Symbol 的字元放入[…]，判斷屬於 Symbol 的字元會回傳給 parser。

```
symbol [,:;()\[\]{}]
```

圖片 3-1、Regular Expression of Symbol

(二)、Arithmetic, Relational and Logical Operators

將 Operator 先分成後綴是否可以將上「=」者，不行者使用|連接，可以者又可以分成單一字元或多個字元，多個字元者包括「>>>」、「>>」、「<<」三種。測試結果可以參考五、執行結果(四)。

```
operator ~|&&|\|\||\+\+|\-\-|([\+\-\*\/%\^\|\&<>=!]|[>]?>>|<<)[=]?
```

圖片 3-2、Regular Expression of Operator

(三)、Reserved words

將每個保留字使用「|」連接即可。

```
reserved_word abstract|assert|boolean|break|byte|case|char|catch|class|const|
continue|default|do|double|else|enum|extends|exxports|false|final|finally|float|
for|if|implements|int|import|instanceof|interface|long|module|main|native|new|
null|print|println|package|private|protected|public|requires|return|short|static|
strictfp|string|String|super|switch|synchronized|this|true|try|throw|throws|
transient|volatile|var|void|volatile|while
```

圖片 3-3、Regular Expression of Reserved words

(四)、Identifiers

Java 的 Identifiers 允許使用「$」、「_」及字母作為開頭，其後可以接上「$」、「_」、數字或字母。

```
id [\$_A-Za-z]+[\$_A-Za-z0-9]*
```

圖片 3-4-1、Regular Expression of Identifiers

Invalid Identifiers 包括使用「^」、「#」及數字作為開頭，其後街上除了數字以外的合法字元，排除數字的原因是可能會遇到「^int」的情形，這種情況需要將「^」判為 operator。

```
errorid [#^0-9]+[\$_A-Za-z]+
```

圖片 3-4-2、Regular Expression of Invalid Indentifiers

(五)、Integer Constant

分成十六進位與十進位兩種表示方法，十六進位包含數字及 A-F、a-f 的字母，十進位則包括數字。另外如果數字前包括「+」、「-」號的話，就先將包含，再用程式判斷為運算子還正負號(詳細於四、遇到問題討論)。

```
int (([\+\-][ ]*)?0[xX])[0-9A-Fa-f]+|([\+\-][ ]*)?[0-9]+
```

圖片 3-5、Regular Expression of Interger Constant

(六)、Float Constant

Float 可以分成底數與指數兩個部分，底數部分分別有「數字.」、「.數字」、「數字」三種表示方法(「數字.數字」包含在前兩種)，底數的正負號則和 Integer 相同；指數的部分標示方法為有「Ee」+「正負」+「數字」，「正負」的部分可以省略。另外底數及最後的「Ff」可以選擇是否加上。儘管「4」符合這個 Regular Expression 的表示方式，但也符合前一個 Integer 的表示方式，所以會先被前一個 Regular Expression 判斷。

```
float ([\+\-][ ]*)?([0-9]+[\.][0-9]*|[0-9]*[\.][0-9]+|[0-9]+)([Ee][\+\-]?[0-9]+)?[Ff]?
```

圖片 3-6、Regular Expression of Float Constant

以上兩個部份的測試結果可以參考五、測試結果(五)。

(七)、String Constants

Sting 的前後由「”」組成，中間的部分不能出現沒有加上逃脫字元的「”」、「換行」、「\」，但可以出現「\\」、「\”」、「\b」、「\f」、「\n」、「\r」、「\t」、「\0」。

```
string \"([^\r\n\\\"]|([\\][\\\"bfnrt0]))*\"
```

圖片 3-7-1、Regular Expression of String Constant

Invalid String 則可能有只有一邊「\”」，或包括單獨的「\」、「”」兩種情況。儘管有許多合法的 String 也符合這樣的 Regular Expression，但他們在前一項就會被判斷為 String，不會被這個判斷影響。

```
errorstring (\"([^\r\n\"]*)|(\"([\\][\\A-Za-z]))*\")
```

圖片 3-7-1、Regular Expression of Invalid String Constant

(八)、Character Constants

和 String Constants 很像，差別只有開頭「’」以及內容只能出現一次。

```
char \'([^\n\\\']|([\\][\\\'bfnrt0]))\'
```

圖片 3-8-1、Regular Expression of Invalid Character Constant

Invalid Character Constant 則是可能在「'」之間出現多個字或是在逃脫字元「\」後加上除了 bfnrt0 之外的字。

```
errorchar \'([^\n]|([\\][\\\'a-z]))*\'
```

圖片 3-8-1、Regular Expression of Invalid Character Constant

以上兩個部分的測試資料可以參考五、測試結果(六)。

(九)、Comment

Comment 可以分成單行註解或多行註解。單行註解為開頭「\\」，其後加上任意字元直到換行。

```
singlecomment \/\/[^\r\n]*
```

圖片 3-9-1、Regular Expression of Single line Comment

多行註解我選擇使用程式的方式判斷，因此 Regular Expression 只有標示出開頭「/*」及「*/」。(詳細於四、遇到問題討論)。

```
multiplecommentstart \/\*
multiplecommentend \*\/
```

圖片 3-9-1、Regular Expression of Single line Comment

(十)、Symbol Tables

我使用 char** symbolTable(pointer to pointer)來表示 Symbol Table 的二維陣列，第一維代表第幾個 Symbol，第二維代表每個 Symbol 所存的字串。每個存入的 Identifier 以第一維所在位置作為 Index。

create()先將 symbolTable 指標指向 NULL。

lookup()則透過 strcmp 比對傳入的字串是否有在 symbol table 之中，若有的畫則將 index 回傳，若無則回傳-1。

insert()則透過 lookup()確認 symbol table 是否有該字串，沒有的話則透過 realloc 將 symbolTable 增加一個指標的位置，再透過 malloc 及 strcpy 將傳入字串複製到 symbolTable 中多出來指標所指向的位置。

lookup()則將 SymbolTable 中的內容輸出。

除了以上四個 function 之外，我還將上了 release()來釋放 SymbolTable 所佔的記憶體空間。

四、遇到的問題

有關撰寫 Regular Expression 所遇到的問題與狀況都已經於第三部分提及，因此這個部分將以 Regular Expression 以外的問題為主。

(一)、如何判斷「+」、「-」為運算子或數字的正負號?

前面透過 Regular Expression 的判斷方式來將 Integer 及 Float 前的「+」、

「-」都先放入數字，接下來進入 Integer 及 Float 的執行範圍時再另外判斷「+」、「-」屬於何者。設定一個變數 preIsOperand 來判斷前一個部分是不是一個運算元：如果是一個運算元，則這個「+」、「-」則是屬於運算子；反之則屬於正負號。我判斷 Integer、Float、Identifier、String 屬於運算元，如果遇到這些內容則將 perIsOperand 設定為 true，反之則設為 false；另外我還發現如果測試「(運算元)+ 數字」時，這時「+」會因為「)」被判斷為正負號，所以我在 Symbol block 中加上如果讀到「)」則不用改變 perIsOperand，由再前一次結果來判斷這項內容屬於正負號或運算子。

當「+」、「-」判斷為運算子時，則將其獨立輸出，除此之外，需要特別注意「+」、「-」和數字之間可能會有空格，因此我用 numberspace 來記錄之間有幾個空格，之後輸出數字時可以避免印出數字。

關於這點的測試資料可以參考圖某。

(二)、如何判斷多行註解範圍?

一開始我使用 Regular Expression 如圖 4 來判斷註解，但我發現如果多個多行註解的話，這樣的判斷方式會把全部判斷為一個多行註解(例：/*多行註解 1*/程式片段/*多行註解 2*/會將判斷為一個多行註解)。

```
comment (\/\/[^\r\n]*)|(\/\*(.|\n)*?\*\/)
```

圖片 4、Wrong Regular Expression of Comments

會出現以上原因是因為 Regular Expression 中的「*」是採用「最長配對原則」，會將「最長」的配對片段回傳，因此我嘗試尋找是否有「最短配對」的符號。我發現可以在「*」加上「?」使其為「最短配對」，在 regular expression 的測試網站上測試也正確，但是放到程式執行上並無法成功，原因是 lex 只能採用「最長配對原則」，「最短配對原則」只有在 Perl、vim 等其他編輯器上可以使用。

所以我最後選擇直接用程式的方式來判斷多行註解的範圍。方法設定一個變數 multiplecommentflag 來判斷目前的片段是否屬於多行註解的範圍。由 Regular Expression 先判斷為「/*」，則將 flag 設定為 true，接者透過改變 yytext 的值來輸出多行註解的內容直到讀到「*/」；Regular Expression 判斷為「*/」後則將 flag 設定為 false。因此進入任一 block 時都需要先判斷是否為多行註解的範圍，若屬於多行註解的範圍則不做任何操作。經過測試時發現，「*/」可能存在於單行註解中，所以單行註解的程式中需要另外判斷是否有「*/」，如果有的話則需要將 flag 設為 false。

另外還需要注意因為多行註解中可能行跨多行，因此如果讀到「\n」字元的話需要將行數計數器和字元計數器進行改變。

關於這點的測試資料可以參考五、測試結果(七)。

五、執行結果

(一)、測試檔案 Test1.java

```
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ ./B093040003 < Test1.java
Line: 2, 1st char: 1, "public" is a "Reserved word".
Line: 2, 1st char: 8, "class" is a "Reserved word".
Line: 2, 1st char: 14, "Test1" is an "Identifier".
Line: 2, 1st char: 20, "{" is a "Symbol".
Line: 3, 1st char: 5, "public" is a "Reserved word".
Line: 3, 1st char: 12, "static" is a "Reserved word".
Line: 3, 1st char: 19, "int" is a "Reserved word".
Line: 3, 1st char: 23, "add" is an "Identifier".
Line: 3, 1st char: 26, "(" is a "Symbol".
Line: 3, 1st char: 27, "int" is a "Reserved word".
Line: 3, 1st char: 31, "a" is an "Identifier".
Line: 3, 1st char: 32, "," is a "Symbol".
Line: 3, 1st char: 34, "int" is a "Reserved word".
Line: 3, 1st char: 38, "b" is an "Identifier".
Line: 3, 1st char: 39, ")" is a "Symbol".
Line: 3, 1st char: 41, "{" is a "Symbol".
Line: 4, 1st char: 9, "return" is a "Reserved word".
Line: 4, 1st char: 16, "a" is an "Identifier".
Line: 4, 1st char: 18, "+" is a "Operator".
Line: 4, 1st char: 20, "b" is an "Identifier".
Line: 4, 1st char: 21, ";" is a "Symbol".
Line: 5, 1st char: 5, "}" is a "Symbol".
Line: 7, 1st char: 5, "public" is a "Reserved word".
Line: 7, 1st char: 12, "static" is a "Reserved word".
Line: 7, 1st char: 19, "void" is a "Reserved word".
Line: 7, 1st char: 24, "main" is a "Reserved word".
Line: 7, 1st char: 28, "(" is a "Symbol".
Line: 7, 1st char: 29, ")" is a "Symbol".
Line: 7, 1st char: 31, "{" is a "Symbol".
Line: 9, 1st char: 9, "int" is a "Reserved word".
Line: 9, 1st char: 13, "c" is an "Identifier".
Line: 9, 1st char: 14, ";" is a "Symbol".
Line: 10, 1st char: 9, "int" is a "Reserved word".
Line: 10, 1st char: 13, "a" is an "Identifier".
Line: 10, 1st char: 15, "=" is a "Operator".
Line: 10, 1st char: 17, "5" is an "Interger Constant".
Line: 10. 1st char: 18. ";" is a "Symbol".
Line: 11, 1st char: 9, "c" is an "Identifier".
Line: 11, 1st char: 11, "=" is a "Operator".
Line: 11, 1st char: 13, "add" is an "Identifier".
Line: 11, 1st char: 16, "(" is a "Symbol".
Line: 11, 1st char: 17, "a" is an "Identifier".
Line: 11, 1st char: 18, "," is a "Symbol".
Line: 11, 1st char: 20, "10" is an "Interger Constant".
Line: 11, 1st char: 22, ")" is a "Symbol".
Line: 11, 1st char: 23, ";" is a "Symbol".
Line: 12, 1st char: 9, "if" is a "Reserved word".
Line: 12, 1st char: 12, "(" is a "Symbol".
Line: 12, 1st char: 13, "c" is an "Identifier".
Line: 12, 1st char: 15, ">" is a "Operator".
Line: 12, 1st char: 17, "10" is an "Interger Constant".
Line: 12, 1st char: 19, ")" is a "Symbol".
Line: 13, 1st char: 13, "print" is a "Reserved word".
Line: 13, 1st char: 18, "(" is a "Symbol".
Line: 13, 1st char: 20, "c = " is a "String".
Line: 13, 1st char: 26, "+" is a "Operator".
Line: 13, 1st char: 28, "-" is a "Operator".
Line: 13, 1st char: 29, "c" is an "Identifier".
Line: 13, 1st char: 30, ")" is a "Symbol".
Line: 13, 1st char: 31, ";" is a "Symbol".
Line: 14, 1st char: 9, "else" is a "Reserved word".
Line: 15, 1st char: 13, "print" is a "Reserved word".
Line: 15, 1st char: 18, "(" is a "Symbol".
Line: 15, 1st char: 19, "c" is an "Identifier".
Line: 15, 1st char: 20, ")" is a "Symbol".
Line: 15, 1st char: 21, ";" is a "Symbol".
Line: 16, 1st char: 9, "print" is a "Reserved word".
Line: 16, 1st char: 14, "(" is a "Symbol".
Line: 16, 1st char: 16, "Hello World" is a "String".
Line: 16, 1st char: 28, ")" is a "Symbol".
Line: 16, 1st char: 29, ";" is a "Symbol".
Line: 18, 1st char: 5, "}" is a "Symbol".
Line: 20, 1st char: 1, "}" is a "Symbol".
The symbol table contains:
Test1
add
a
b
c
```

圖片 5-1、Test1 執行結果

(二)、測試檔案 Test2.java

```
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ ./B093040003 < Test2.java
Line: 1, 1st char: 1, // this is a comment // line */ /* with /* delimiters */ before the end is a "Single line Comment".
Line: 3, 1st char: 1, "public" is a "Reserved word".
Line: 3, 1st char: 8, "class" is a "Reserved word".
Line: 3, 1st char: 14, "Test2" is an "Identifier".
Line: 3, 1st char: 20, "{" is a "Symbol".
Line: 4, 1st char: 5, "int" is a "Reserved word".
Line: 4, 1st char: 9, "i" is an "Identifier".
Line: 4, 1st char: 11, "=" is a "Operator".
Line: 4, 1st char: 13, "-100" is an "Interger Constant".
Line: 4, 1st char: 17, ";" is a "Symbol".
Line: 5, 1st char: 5, "double" is a "Reserved word".
Line: 5, 1st char: 12, "d" is an "Identifier".
Line: 5, 1st char: 14, "=" is a "Operator".
Line: 5, 1st char: 16, "12.25e+6" is a "Float Constant".
Line: 5, 1st char: 24, ";" is a "Symbol".
Line: 7, 1st char: 5, "public" is a "Reserved word".
Line: 7, 1st char: 12, "static" is a "Reserved word".
Line: 7, 1st char: 19, "void" is a "Reserved word".
Line: 7, 1st char: 24, "main" is a "Reserved word".
Line: 7, 1st char: 28, "(" is a "Symbol".
Line: 7, 1st char: 29, ")" is a "Symbol".
Line: 7, 1st char: 31, "{" is a "Symbol".
Line: 8, 1st char: 1,
/*this is a comment // line with some /* and
// delimiters */
is a "multiple line Comment".
Line: 10, 1st char: 5, "}" is a "Symbol".
Line: 11, 1st char: 1, "}" is a "Symbol".
The symbol table contains:
Test2
i
d
```

圖片 5-2、Test2 執行結果

(三)、測試檔案 Test3.java

```
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ ./B093040003 < Test3.java
Line: 2, 1st char: 1, "public" is a "Reserved word".
Line: 2, 1st char: 8, "class" is a "Reserved word".
Line: 2, 1st char: 14, "Test3" is an "Identifier".
Line: 2, 1st char: 20, "{" is a "Symbol".
Line: 3, 1st char: 5, "int" is a "Reserved word".
Line: 3, 1st char: 9, "A" is an "Identifier".
Line: 3, 1st char: 10, ";" is a "Symbol".
Line: 4, 1st char: 5, "int" is a "Reserved word".
Line: 4, 1st char: 9, "a" is an "Identifier".
Line: 5, 1st char: 5, "double" is a "Reserved word".
Line: 5, 1st char: 12, "b" is an "Identifier".
Line: 5, 1st char: 13, ";" is a "Symbol".
Line: 6, 1st char: 5, "double" is a "Reserved word".
Line: 6, 1st char: 12, "A" is an "Identifier".
Line: 6, 1st char: 13, ";" is a "Symbol".
Line: 8, 1st char: 5, "public" is a "Reserved word".
Line: 8, 1st char: 12, "Test3" is an "Identifier".
Line: 8, 1st char: 17, "(" is a "Symbol".
Line: 8, 1st char: 18, ")" is a "Symbol".
Line: 8, 1st char: 20, "{" is a "Symbol".
Line: 9, 1st char: 9, "a" is an "Identifier".
Line: 9, 1st char: 11, "=" is a "Operator".
Line: 9, 1st char: 13, "1" is an "Interger Constant".
Line: 9, 1st char: 14, ";" is a "Symbol".
Line: 10, 1st char: 9, "A" is an "Identifier".
Line: 10, 1st char: 11, "=" is a "Operator".
Line: 10, 1st char: 13, "2" is an "Interger Constant".
Line: 10, 1st char: 14, ";" is a "Symbol".
Line: 11, 1st char: 9, "b" is an "Identifier".
Line: 11, 1st char: 11, "=" is a "Operator".
Line: 11, 1st char: 13, "-1.2" is a "Float Constant".
Line: 11, 1st char: 17, ";" is a "Symbol".
Line: 12, 1st char: 5, "}" is a "Symbol".
Line: 13, 1st char: 1, "}" is a "Symbol".
The symbol table contains:
Test3
A
a
b
```

圖片 5-3、Test3 執行結果

(四)、測試檔案 operatortest.java

```
1 ~ && || ++ --
2 + - * / % ^ | & < > = ! >> >>> <<
3 += -= *= /= %= ^= |= &= <= >= == != >>= >>>= <<=
4
5 ~= &&= ||= ++= --= =~ =& =| =!
6 +++---+--++
7 >>>>>>>>>>>>>>>>
```

圖片 5-4-1、operatortest.java 內容

```
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ ./B093040003 < operatortest.java
Line: 1, 1st char: 1, "~" is a "Operator".
Line: 1, 1st char: 3, "&&" is a "Operator".
Line: 1, 1st char: 6, "||" is a "Operator".
Line: 1, 1st char: 9, "++" is a "Operator".
Line: 1, 1st char: 12, "--" is a "Operator".
Line: 2, 1st char: 1, "+" is a "Operator".
Line: 2, 1st char: 3, "-" is a "Operator".
Line: 2, 1st char: 5, "*" is a "Operator".
Line: 2, 1st char: 7, "/" is a "Operator".
Line: 2, 1st char: 9, "%" is a "Operator".
Line: 2, 1st char: 11, "^" is a "Operator".
Line: 2, 1st char: 13, "|" is a "Operator".
Line: 2, 1st char: 15, "&" is a "Operator".
Line: 2, 1st char: 17, "<" is a "Operator".
Line: 2, 1st char: 19, ">" is a "Operator".
Line: 2, 1st char: 21, "=" is a "Operator".
Line: 2, 1st char: 23, "!" is a "Operator".
Line: 2, 1st char: 25, ">>" is a "Operator".
Line: 2, 1st char: 28, ">>>" is a "Operator".
Line: 2, 1st char: 32, "<<" is a "Operator".
Line: 3, 1st char: 1, "+=" is a "Operator".
Line: 3, 1st char: 4, "-=" is a "Operator".
Line: 3, 1st char: 7, "*=" is a "Operator".
Line: 3, 1st char: 10, "/=" is a "Operator".
Line: 3, 1st char: 13, "%=" is a "Operator".
Line: 3, 1st char: 16, "^=" is a "Operator".
Line: 3, 1st char: 19, "|=" is a "Operator".
Line: 3, 1st char: 22, "&=" is a "Operator".
Line: 3, 1st char: 25, "<=" is a "Operator".
Line: 3, 1st char: 28, ">=" is a "Operator".
Line: 3, 1st char: 31, "==" is a "Operator".
Line: 3, 1st char: 34, "!=" is a "Operator".
Line: 3, 1st char: 37, ">>=" is a "Operator".
Line: 3, 1st char: 41, ">>>=" is a "Operator".
Line: 3, 1st char: 46, "<<=" is a "Operator".
Line: 5, 1st char: 1, "~" is a "Operator".
Line: 5, 1st char: 2, "=" is a "Operator".
Line: 5, 1st char: 4, "&&" is a "Operator".
Line: 5, 1st char: 6, "=" is a "Operator".
Line: 5, 1st char: 8, "||" is a "Operator".
Line: 5, 1st char: 10, "=" is a "Operator".
Line: 5, 1st char: 12, "++" is a "Operator".
Line: 5, 1st char: 14, "=" is a "Operator".
Line: 5, 1st char: 16, "--" is a "Operator".
Line: 5, 1st char: 18, "=" is a "Operator".
Line: 5, 1st char: 20, "=" is a "Operator".
Line: 5, 1st char: 21, "~" is a "Operator".
Line: 5, 1st char: 23, "=" is a "Operator".
Line: 5, 1st char: 24, "&" is a "Operator".
Line: 5, 1st char: 26, "=" is a "Operator".
Line: 5, 1st char: 27, "|" is a "Operator".
Line: 5, 1st char: 29, "=" is a "Operator".
Line: 5, 1st char: 30, "!" is a "Operator".
Line: 6, 1st char: 1, "++" is a "Operator".
Line: 6, 1st char: 3, "+" is a "Operator".
Line: 6, 1st char: 4, "--" is a "Operator".
Line: 6, 1st char: 6, "-" is a "Operator".
Line: 6, 1st char: 7, "+" is a "Operator".
Line: 6, 1st char: 8, "--" is a "Operator".
Line: 6, 1st char: 10, "++" is a "Operator".
Line: 7, 1st char: 1, ">>>" is a "Operator".
Line: 7, 1st char: 4, ">>>" is a "Operator".
Line: 7, 1st char: 7, ">>" is a "Operator".
Line: 7, 1st char: 9, "<<" is a "Operator".
Line: 7, 1st char: 11, "<<" is a "Operator".
Line: 7, 1st char: 13, "<<" is a "Operator".
Line: 7, 1st char: 15, "<<" is a "Operator".
Line: 7, 1st char: 17, "<" is a "Operator".
The symbol table contains: zero item
```

圖片 5-4-2、operatortest.java 測試結果

(五)、測試檔案 signnumbertest.java

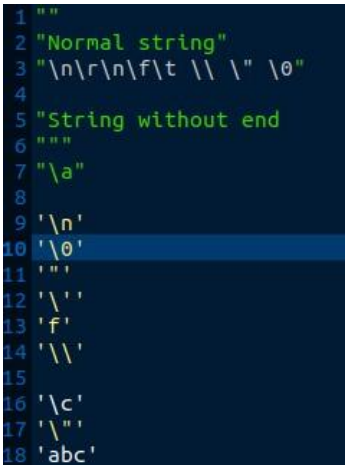
```
0xA19f - 0x23f3
+123 + 534
-    2939    +      384
-   - 23-+39
3+4
3e--3
-3.14 3. -.14 3.14f 3.f .14f
+3.14E16 3.e-7 -.14e+12
+3.14E16f 3.e-7f .14e+12f
432f 3e-4 4E12f
(+  7E-2 + -7.5E+3) -7.5E+3f
```

圖片 5-5-2、測試檔案 signnumbertest.java 內容

```
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ ./B093040003 < signnumbertest.java
Line: 1, 1st char: 1, "0xA19f" is an "Interger Constant".
Line: 1, 1st char: 8, "-" is an "Operator".
Line: 1, 1st char: 10, "0x23f3" is an "Interger Constant".
Line: 2, 1st char: 1, "+123" is an "Interger Constant".
Line: 2, 1st char: 6, "+" is an "Operator".
Line: 2, 1st char: 8, "534" is an "Interger Constant".
Line: 3, 1st char: 1, "-    2939" is an "Interger Constant".
Line: 3, 1st char: 14, "+" is an "Operator".
Line: 3, 1st char: 21, "384" is an "Interger Constant".
Line: 4, 1st char: 1, "-" is a "Operator".
Line: 4, 1st char: 5, "- 23" is an "Interger Constant".
Line: 4, 1st char: 9, "-" is a "Operator".
Line: 4, 1st char: 10, "+39" is an "Interger Constant".
Line: 5, 1st char: 1, "3" is an "Interger Constant".
Line: 5, 1st char: 2, "+" is an "Operator".
Line: 5, 1st char: 3, "4" is an "Interger Constant".
Line: 6, 1st char: 1, "3e" is an "Invaild Identifier".
Line: 6, 1st char: 3, "--" is a "Operator".
Line: 6, 1st char: 5, "3" is an "Interger Constant".
Line: 7, 1st char: 1, "-3.14" is a "Float Constant".
Line: 7, 1st char: 7, "3." is a "Float Constant".
Line: 7, 1st char: 10, "-" is an "Operator".
Line: 7, 1st char: 11, ".14" is a "Float Constant".
Line: 7, 1st char: 15, "3.14f" is a "Float Constant".
Line: 7, 1st char: 21, "3.f" is a "Float Constant".
Line: 7, 1st char: 25, ".14f" is a "Float Constant".
Line: 8, 1st char: 1, "+3.14E16" is a "Float Constant".
Line: 8, 1st char: 10, "3.e-7" is a "Float Constant".
Line: 8, 1st char: 16, "-" is an "Operator".
Line: 8, 1st char: 17, ".14e+12" is a "Float Constant".
Line: 9, 1st char: 1, "+3.14E16f" is a "Float Constant".
Line: 9, 1st char: 11, "3.e-7f" is a "Float Constant".
Line: 9, 1st char: 18, ".14e+12f" is a "Float Constant".
Line: 10, 1st char: 1, "432f" is a "Float Constant".
Line: 10, 1st char: 6, "3e-4" is a "Float Constant".
Line: 10, 1st char: 11, "4E12f" is a "Float Constant".
Line: 11, 1st char: 1, "(" is a "Symbol".
Line: 11, 1st char: 2, "+  7E-2" is a "Float Constant".
Line: 11, 1st char: 10, "+" is a "Operator".
Line: 11, 1st char: 12, "-7.5E+3" is a "Float Constant".
Line: 11, 1st char: 19, ")" is a "Symbol".
Line: 11, 1st char: 21, "-" is an "Operator".
Line: 11, 1st char: 22, "7.5E+3f" is a "Float Constant".
The symbol table contains: zero item
```

圖片 5-5-2、signnumbertest.java 測試結果

(六)、測試檔案 stringchartest.java



圖片 5-6-1、stringchartest.java 內容

```
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ ./B093040003 < stringchartest.java
Line: 1, 1st char: 2, "" is a "String".
Line: 2, 1st char: 2, "Normal string" is a "String".
Line: 3, 1st char: 2, "\n\r\n\f\t \\ \" \0" is a "String".
Line: 5, 1st char: 1, "String without end is an "Invaild String".
Line: 6, 1st char: 2, "" is a "String".
Line: 6, 1st char: 3, " is an "Invaild String".
Line: 7, 1st char: 1, "\a" is an "Invaild String".
Line: 9, 1st char: 1, '\n' is a "Character".
Line: 10, 1st char: 1, '\0' is a "Character".
Line: 11, 1st char: 1, '"' is a "Character".
Line: 12, 1st char: 1, '\'' is a "Character".
Line: 13, 1st char: 1, 'f' is a "Character".
Line: 14, 1st char: 1, '\\' is a "Character".
Line: 16, 1st char: 1, '\c' is an "Invaild Character".
Line: 17, 1st char: 1, '\"' is an "Invaild Character".
Line: 18, 1st char: 1, 'abc' is an "Invaild Character".
The symbol table contains: zero item
```

圖片 5-6-1、stringchartest.java 測試結果

(七)、測試檔案 commenttest.java

圖片 5-7-1、commenttest.java 內容

```
ubuntu@ubuntu-virtual-machine:~/Desktop/Compiler/HW1/DemoFile$ ./B093040003 < commenttest.java
Line: 1, 1st char: 1, // one line comment is a "Single line Comment".
Line: 3, 1st char: 1, //one line comment with /* */ is a "Single line Comment".
Line: 6, 1st char: 1,
/**
multiple comment
// wiht one line comment
***** /***
//////////////
*/
is a "multiple line Comment".
Line: 13, 1st char: 1,
/*another multiple
*/
is a "multiple line Comment".
Line: 17, 1st char: 1, // this is a "Single line Comment".
The symbol table contains: zero item
```

圖片 5-7-1、commenttest.java 測試結果