Readme

學號：B073021024

姓名：　　　錢承

1. Lex 版本

   Flex 2.6.4

2. 作業平台

   Ubuntu 20.04.2 LTS （Linux 5.8.0-49-generic）

3. 執行方式

   Bash Terminal 輸入以下指令：

   無匯入檔案　：make ; ./demo

   匯入檔案　　：make ; ./demo < 檔案名稱

4. 處理規格書上的問題

   Symbols：

   直接選取該符號，如 Comma: \, 。

   Arithmetic, Relational, and Logical Operators：

   直接選取該符號，如 multiplication : * 。

   Reserved words：

   直接將該保留字寫出，如 boolean|break|……。

   Identifiers：

   分為合法和不合法兩個部分。

   合法：

   選取大小寫 a 至 z 開頭或"_"開頭，並且後面為零到多個大小寫 a 至 z 開頭或數字或"_"。

   不合法：

   選取數字或"^","#"開頭。

   Integer Constants：

   選取零或多個"+"或"-"開頭，一或多個數字結尾。

   Float Constants：

   分為非科學記號和科學記號。

   非科學記號：

   選取零或多個"+"或"-"開頭，然後選取一或多個數字，再來選取".", 最後選取一或多個數字。

科學記號：

選取零或多個”+”或”-”開頭，然後選取一或多個數字，再來選取”.”, 再選取 e 或 E，最後選取數字。

String Constants：

選取” “ ”，然後選取零到多個不包含” ’ ”,” \ ”,” “ ”或包含”\”且後面包含” “ ”,” ‘ ”,” \ ”,” t ”,” r ”,” n ”,” b ”,” f ”，最後選取” “ ”。

Whitespace：

直接選取 tab 或空白，換行包含\n、\r 其中之一，或同時出現。

Comments：

分為 C type 和 C++ type。

C type:

選取”/*” 開頭，選取零到多個不包含” * ”,再選取一到多個” * ”,再來重複執行後面，選取一個不包含” / ”，後面接不包含” * ”，最後選取包含一到多個” * ”。

C++ type:

選取”//”開頭並且換行前所有字元。

Symbol table:

利用指標實作類似 C++中 vector 的功能，可以儲存字串，並且動態調整陣列大小。

5. 作業所遇到的問題

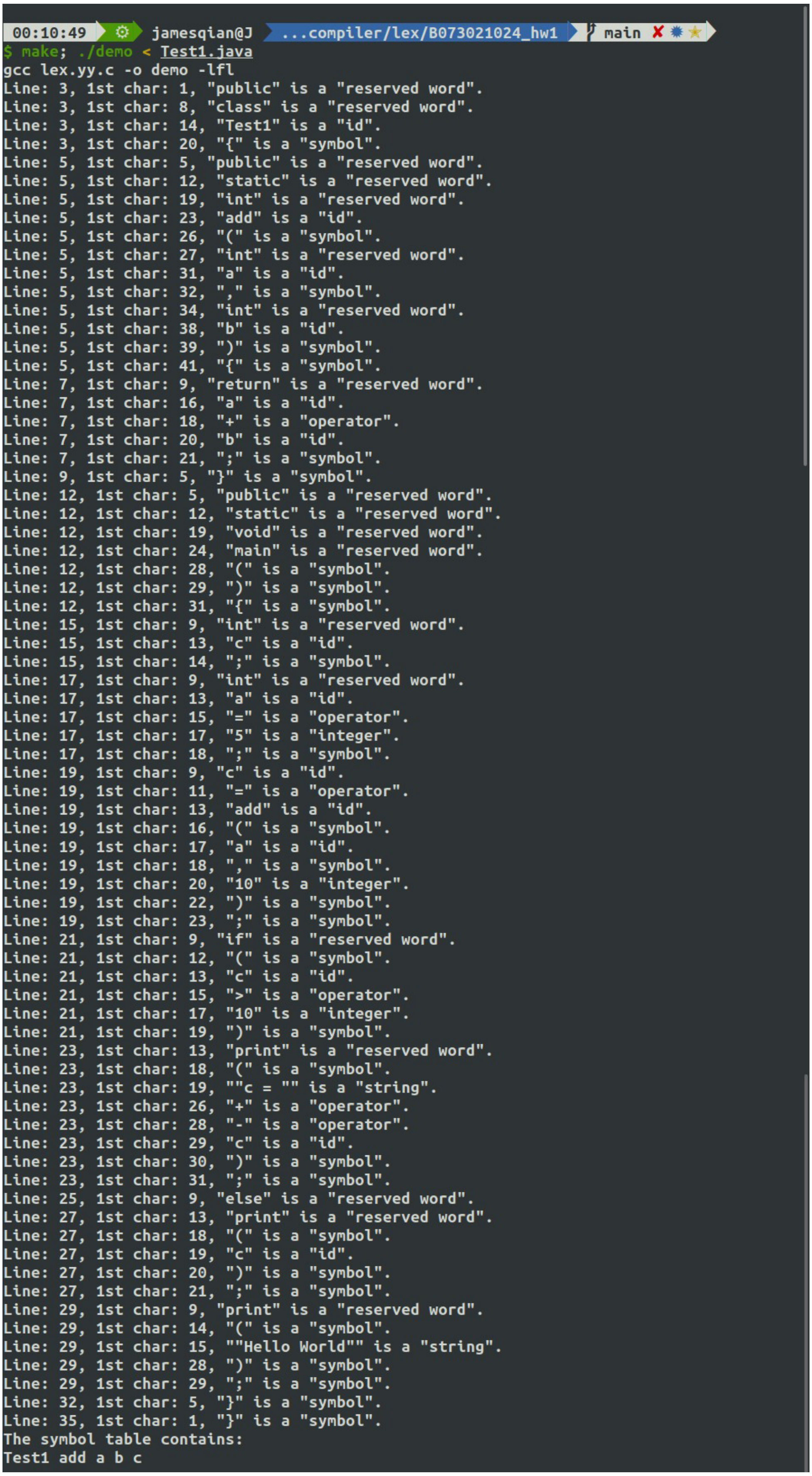
(1) 換行不只包含\n，而且還包含\r。

(2) 在處理 comment 時，必須特別處理其中的換行，因為裡面的換行不會被 eol match 到。

(3) C 語言不能儲存 string，而且不支援 vector，因此 symbol table 製作較為複雜。

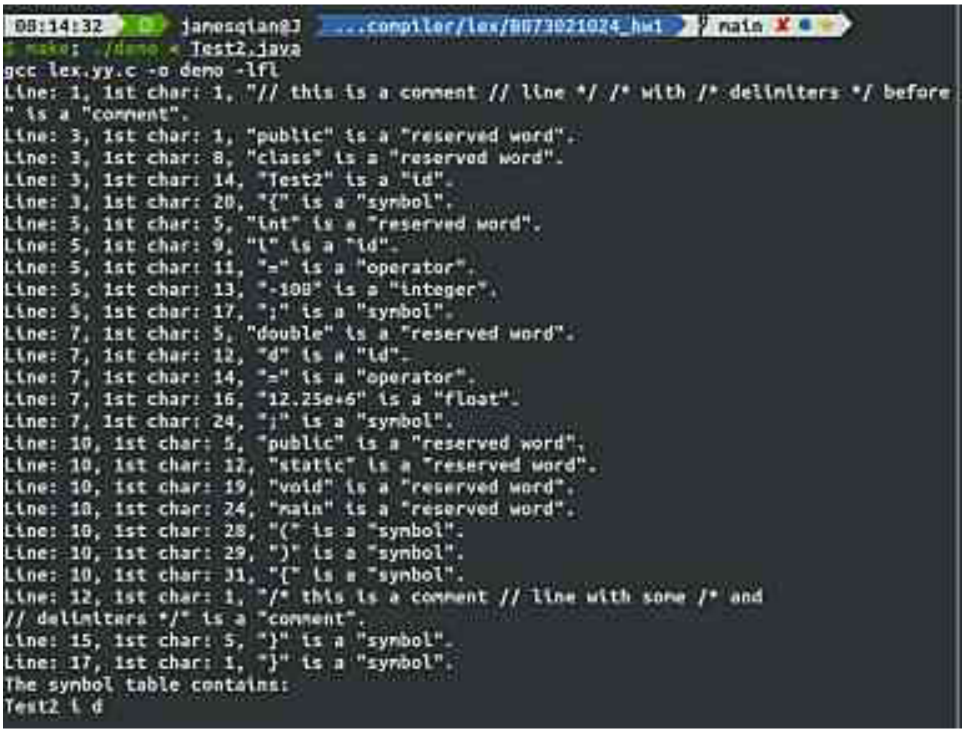
6. 所有測試檔執行出來的結果

(1) Test1.java



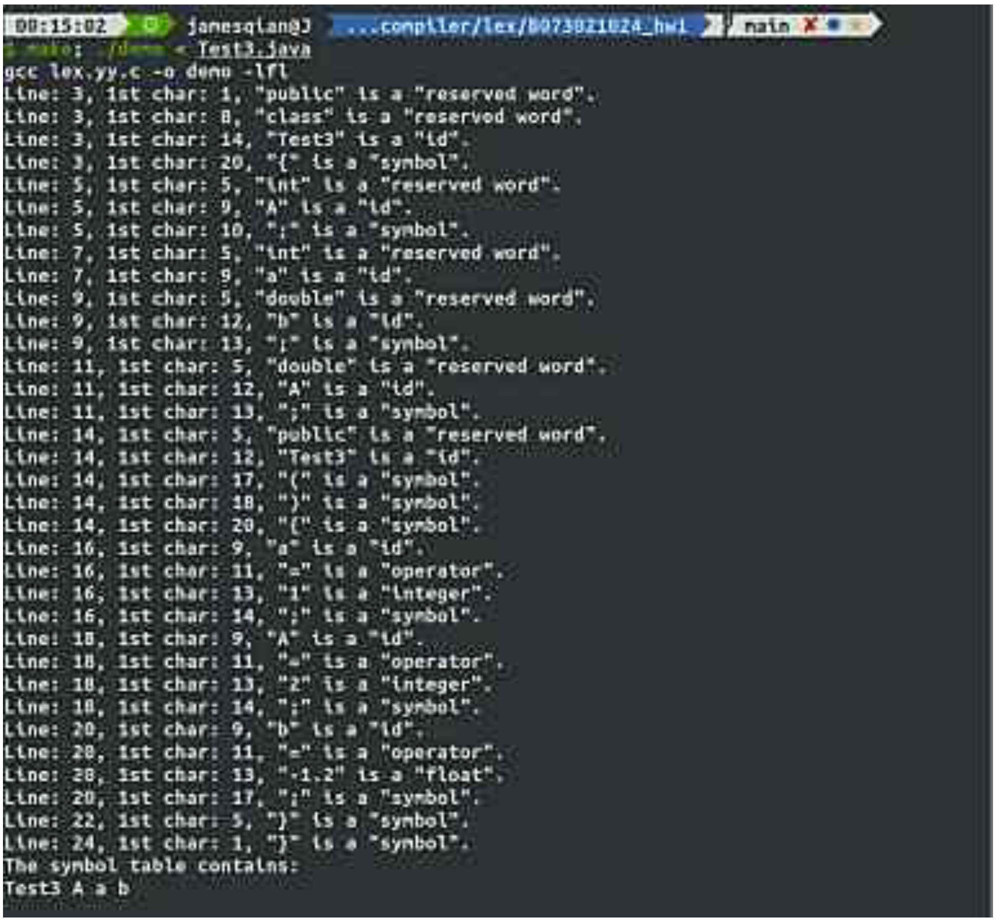
```
00:10:49 ⚙ jamesqian@J ...compiler/lex/B073021024_hw1 main ✘ ✹ ★
$ make; ./demo < Test1.java
gcc lex.yy.c -o demo -lfl
Line: 3, 1st char: 1, "public" is a "reserved word".
Line: 3, 1st char: 8, "class" is a "reserved word".
Line: 3, 1st char: 14, "Test1" is a "id".
Line: 3, 1st char: 20, "{" is a "symbol".
Line: 5, 1st char: 5, "public" is a "reserved word".
Line: 5, 1st char: 12, "static" is a "reserved word".
Line: 5, 1st char: 19, "int" is a "reserved word".
Line: 5, 1st char: 23, "add" is a "id".
Line: 5, 1st char: 26, "(" is a "symbol".
Line: 5, 1st char: 27, "int" is a "reserved word".
Line: 5, 1st char: 31, "a" is a "id".
Line: 5, 1st char: 32, "," is a "symbol".
Line: 5, 1st char: 34, "int" is a "reserved word".
Line: 5, 1st char: 38, "b" is a "id".
Line: 5, 1st char: 39, ")" is a "symbol".
Line: 5, 1st char: 41, "{" is a "symbol".
Line: 7, 1st char: 9, "return" is a "reserved word".
Line: 7, 1st char: 16, "a" is a "id".
Line: 7, 1st char: 18, "+" is a "operator".
Line: 7, 1st char: 20, "b" is a "id".
Line: 7, 1st char: 21, ";" is a "symbol".
Line: 9, 1st char: 5, "}" is a "symbol".
Line: 12, 1st char: 5, "public" is a "reserved word".
Line: 12, 1st char: 12, "static" is a "reserved word".
Line: 12, 1st char: 19, "void" is a "reserved word".
Line: 12, 1st char: 24, "main" is a "reserved word".
Line: 12, 1st char: 28, "(" is a "symbol".
Line: 12, 1st char: 29, ")" is a "symbol".
Line: 12, 1st char: 31, "{" is a "symbol".
Line: 15, 1st char: 9, "int" is a "reserved word".
Line: 15, 1st char: 13, "c" is a "id".
Line: 15, 1st char: 14, ";" is a "symbol".
Line: 17, 1st char: 9, "int" is a "reserved word".
Line: 17, 1st char: 13, "a" is a "id".
Line: 17, 1st char: 15, "=" is a "operator".
Line: 17, 1st char: 17, "5" is a "integer".
Line: 17, 1st char: 18, ";" is a "symbol".
Line: 19, 1st char: 9, "c" is a "id".
Line: 19, 1st char: 11, "=" is a "operator".
Line: 19, 1st char: 13, "add" is a "id".
Line: 19, 1st char: 16, "(" is a "symbol".
Line: 19, 1st char: 17, "a" is a "id".
Line: 19, 1st char: 18, "," is a "symbol".
Line: 19, 1st char: 20, "10" is a "integer".
Line: 19, 1st char: 22, ")" is a "symbol".
Line: 19, 1st char: 23, ";" is a "symbol".
Line: 21, 1st char: 9, "if" is a "reserved word".
Line: 21, 1st char: 12, "(" is a "symbol".
Line: 21, 1st char: 13, "c" is a "id".
Line: 21, 1st char: 15, ">" is a "operator".
Line: 21, 1st char: 17, "10" is a "integer".
Line: 21, 1st char: 19, ")" is a "symbol".
Line: 23, 1st char: 13, "print" is a "reserved word".
Line: 23, 1st char: 18, "(" is a "symbol".
Line: 23, 1st char: 19, ""c = "" is a "string".
Line: 23, 1st char: 26, "+" is a "operator".
Line: 23, 1st char: 28, "-" is a "operator".
Line: 23, 1st char: 29, "c" is a "id".
Line: 23, 1st char: 30, ")" is a "symbol".
Line: 23, 1st char: 31, ";" is a "symbol".
Line: 25, 1st char: 9, "else" is a "reserved word".
Line: 27, 1st char: 13, "print" is a "reserved word".
Line: 27, 1st char: 18, "(" is a "symbol".
Line: 27, 1st char: 19, "c" is a "id".
Line: 27, 1st char: 20, ")" is a "symbol".
Line: 27, 1st char: 21, ";" is a "symbol".
Line: 29, 1st char: 9, "print" is a "reserved word".
Line: 29, 1st char: 14, "(" is a "symbol".
Line: 29, 1st char: 15, ""Hello World"" is a "string".
Line: 29, 1st char: 28, ")" is a "symbol".
Line: 29, 1st char: 29, ";" is a "symbol".
Line: 32, 1st char: 5, "}" is a "symbol".
Line: 35, 1st char: 1, "}" is a "symbol".
The symbol table contains:
Test1 add a b c
```

(2) Test2.java



```
08:14:32  jamesqian@J  ...compiler/lex/B073021024_hw1  main
$ make: ./demo < Test2.java
gcc lex.yy.c -o demo -lfl
Line: 1, 1st char: 1, "// this is a comment // line */ /* with /* delimiters */ before
" is a "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 a "id".
Line: 3, 1st char: 20, "{" is a "symbol".
Line: 5, 1st char: 5, "int" is a "reserved word".
Line: 5, 1st char: 9, "i" is a "id".
Line: 5, 1st char: 11, "=" is a "operator".
Line: 5, 1st char: 13, "-108" is a "integer".
Line: 5, 1st char: 17, ";" is a "symbol".
Line: 7, 1st char: 5, "double" is a "reserved word".
Line: 7, 1st char: 12, "d" is a "id".
Line: 7, 1st char: 14, "=" is a "operator".
Line: 7, 1st char: 16, "12.25e+6" is a "float".
Line: 7, 1st char: 24, ";" is a "symbol".
Line: 10, 1st char: 5, "public" is a "reserved word".
Line: 10, 1st char: 12, "static" is a "reserved word".
Line: 10, 1st char: 19, "void" is a "reserved word".
Line: 10, 1st char: 24, "main" is a "reserved word".
Line: 10, 1st char: 28, "(" is a "symbol".
Line: 10, 1st char: 29, ")" is a "symbol".
Line: 10, 1st char: 31, "{" is a "symbol".
Line: 12, 1st char: 1, "/* this is a comment // line with some /* and
// delimiters */" is a "comment".
Line: 15, 1st char: 5, "}" is a "symbol".
Line: 17, 1st char: 1, "}" is a "symbol".
The symbol table contains:
Test2 i d
```

(3) Test3.java



```
08:15:02  jamesqian@J  ...compiler/lex/B073021024_hw1  main
$ make: ./demo < Test3.java
gcc lex.yy.c -o demo -lfl
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, "Test3" is a "id".
Line: 3, 1st char: 20, "{" is a "symbol".
Line: 5, 1st char: 5, "int" is a "reserved word".
Line: 5, 1st char: 9, "A" is a "id".
Line: 5, 1st char: 10, ";" is a "symbol".
Line: 7, 1st char: 5, "int" is a "reserved word".
Line: 7, 1st char: 9, "a" is a "id".
Line: 9, 1st char: 5, "double" is a "reserved word".
Line: 9, 1st char: 12, "b" is a "id".
Line: 9, 1st char: 13, ";" is a "symbol".
Line: 11, 1st char: 5, "double" is a "reserved word".
Line: 11, 1st char: 12, "A" is a "id".
Line: 11, 1st char: 13, ";" is a "symbol".
Line: 14, 1st char: 5, "public" is a "reserved word".
Line: 14, 1st char: 12, "Test3" is a "id".
Line: 14, 1st char: 17, "(" is a "symbol".
Line: 14, 1st char: 18, ")" is a "symbol".
Line: 14, 1st char: 20, "{" is a "symbol".
Line: 16, 1st char: 9, "a" is a "id".
Line: 16, 1st char: 11, "=" is a "operator".
Line: 16, 1st char: 13, "1" is a "integer".
Line: 16, 1st char: 14, ";" is a "symbol".
Line: 18, 1st char: 9, "A" is a "id".
Line: 18, 1st char: 11, "=" is a "operator".
Line: 18, 1st char: 13, "2" is a "integer".
Line: 18, 1st char: 14, ";" is a "symbol".
Line: 20, 1st char: 9, "b" is a "id".
Line: 20, 1st char: 11, "=" is a "operator".
Line: 20, 1st char: 13, "-1.2" is a "float".
Line: 20, 1st char: 17, ";" is a "symbol".
Line: 22, 1st char: 5, "}" is a "symbol".
Line: 24, 1st char: 1, "}" is a "symbol".
The symbol table contains:
Test3 A a b
```