1. Lex 版本: flex 2.6.4
2. 作業平台: Ubuntu 18.04.4
3. 執行方式:
   A. sudo apt-get install flex
   B. 編寫程式碼
   C. make

      (flex B075020033.l

      gcc lex.yy.c -o demo -lfl)
   D. ./demo < Test1.java

      ./demo < Test2.java

      ./demo < Test3.java

4. 你/妳如何處理這份規格書上的問題:
   A. 查 lex 的規則、regular expression、hash table 的用法
   B. Symbol table 選擇用 array 處理
   C. 有些 token 很難定義或是一些 regular expressionn 問題，會去跟同學討論，參考他們的做法去理解
   D. 錯誤訊息的部分，從規格書上面的舉例去實作，我做了三種錯誤訊息:ID、string、我在定義裡沒有寫出來的規則，最後 output 時能在螢幕顯示 invalid id、invalid string、error
   E. 在定義中善用 regular expression 的規則符號:[]\+*?|^等等符號去實作

5. 你/妳寫這個作業所遇到的問題:
   A. 第一次寫 lex，不熟悉語法，超多用法都不知道怎麼做，不斷的上網查詢才拼拼湊湊出程式碼，好不容易完成時卻出現”unrecognized rule”，找到原因是我將 Lex Rule 那邊改格式，結果就出問題了。

EX:

```
{id}{                          {id}
{ //code      我改成→          {
}                              { //code
}                              }
                               }
```

B. 一開始對 regular expression 的用法很陌生，不斷的上網查詢資料及自己推演才得到 token 的寫法，尤其在 string 及 comments 遇到許多困難。

C. 用 hash table 作 symbol table 時，看網路上的範例仍然不理解為什麼要這樣做。因為感覺助教的測資應該不會超大，不考慮演算法問題的話，所以最後決定採用 array。不過也讓我知道自己的演算法與資料結構沒學好，之後還是要再努力複習一下了。

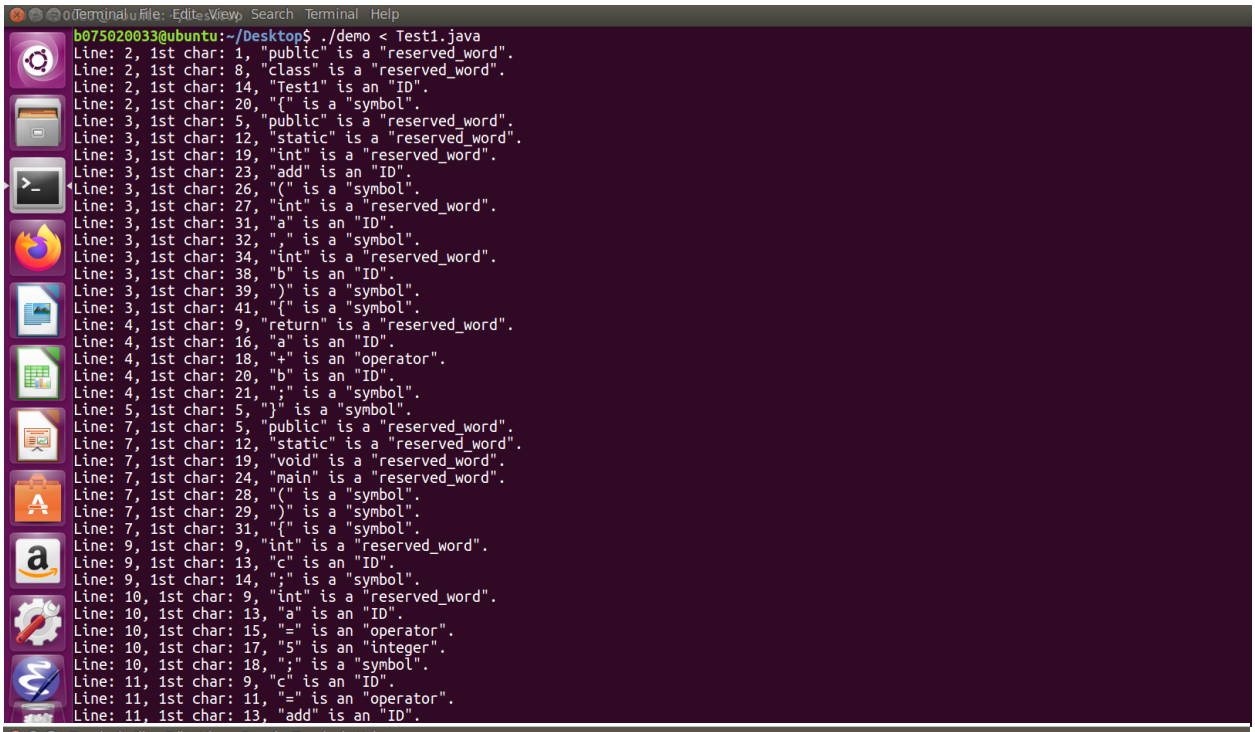
D. 要找出不合法的 token，因為規格書上沒有明講，所以我只能盡量寫，擔心有自己沒想到的情況，可能也會有小 bug。

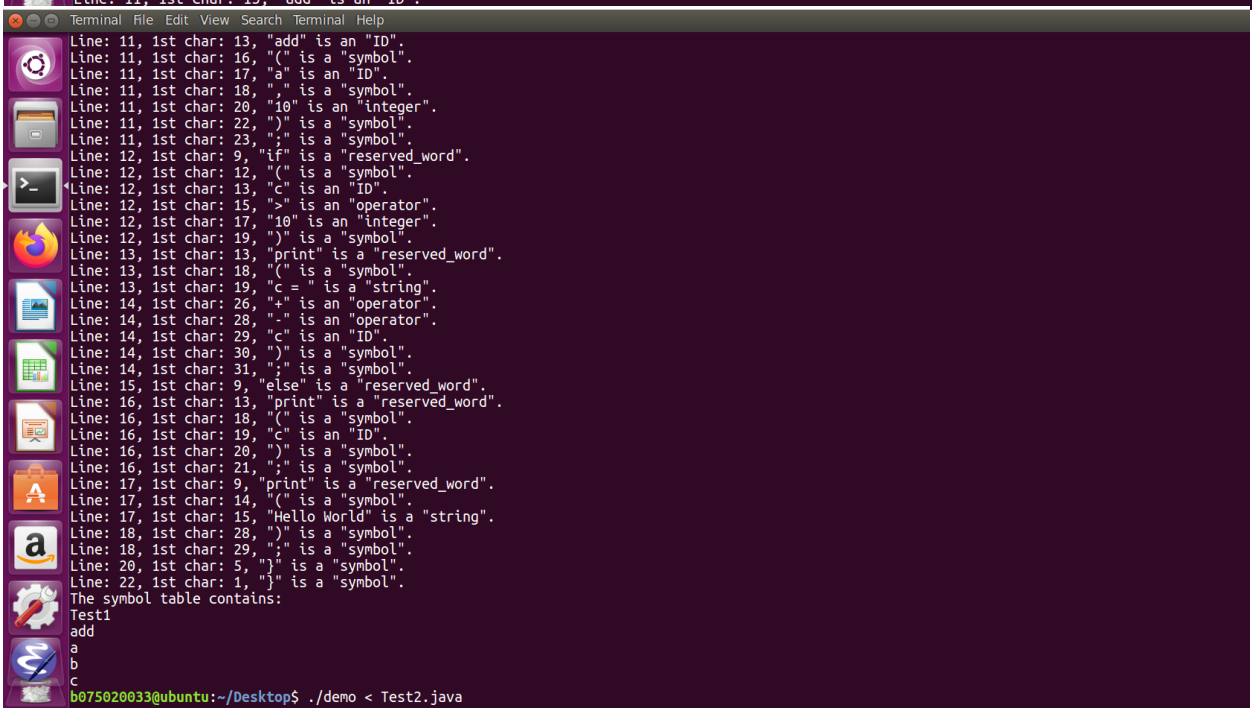
E. 發現每種語法的編譯器相差極大，因為上網查詢有關 JAVA 的資料真的很少(可能我不太會搜尋)，想說參考看看 C 或 C++的範例，發現好多 token 都沒有定義到。

6. 所有測試檔執行出來的結果，存成圖片

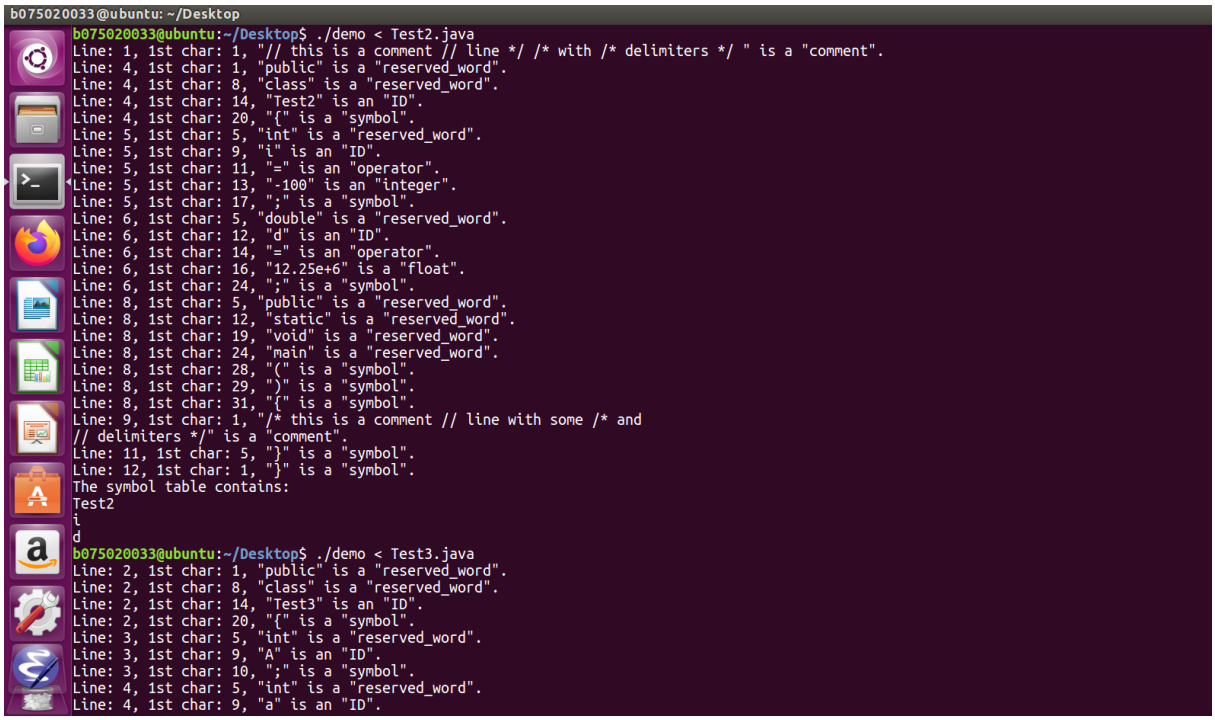
(下一頁)

Test1. java


Terminal File Edit View Search Terminal Help
b075020033@ubuntu:~/Desktop$ ./demo < 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 "ID".
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 "ID".
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 "ID".
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 "ID".
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 "ID".
Line: 4, 1st char: 18, "+" is an "operator".
Line: 4, 1st char: 20, "b" is an "ID".
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 "ID".
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 "ID".
Line: 10, 1st char: 15, "=" is an "operator".
Line: 10, 1st char: 17, "5" is an "integer".
Line: 10, 1st char: 18, ";" is a "symbol".
Line: 11, 1st char: 9, "c" is an "ID".
Line: 11, 1st char: 11, "=" is an "operator".
Line: 11, 1st char: 13, "add" is an "ID".


Terminal File Edit View Search Terminal Help
Line: 11, 1st char: 13, "add" is an "ID".
Line: 11, 1st char: 16, "(" is a "symbol".
Line: 11, 1st char: 17, "a" is an "ID".
Line: 11, 1st char: 18, "," is a "symbol".
Line: 11, 1st char: 20, "10" is an "integer".
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 "ID".
Line: 12, 1st char: 15, ">" is an "operator".
Line: 12, 1st char: 17, "10" is an "integer".
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: 19, "c = " is a "string".
Line: 14, 1st char: 26, "+" is an "operator".
Line: 14, 1st char: 28, "-" is an "operator".
Line: 14, 1st char: 29, "c" is an "ID".
Line: 14, 1st char: 30, ")" is a "symbol".
Line: 14, 1st char: 31, ";" is a "symbol".
Line: 15, 1st char: 9, "else" is a "reserved_word".
Line: 16, 1st char: 13, "print" is a "reserved_word".
Line: 16, 1st char: 18, "(" is a "symbol".
Line: 16, 1st char: 19, "c" is an "ID".
Line: 16, 1st char: 20, ")" is a "symbol".
Line: 16, 1st char: 21, ";" is a "symbol".
Line: 17, 1st char: 9, "print" is a "reserved_word".
Line: 17, 1st char: 14, "(" is a "symbol".
Line: 17, 1st char: 15, "Hello World" is a "string".
Line: 18, 1st char: 28, ")" is a "symbol".
Line: 18, 1st char: 29, ";" is a "symbol".
Line: 20, 1st char: 5, "}" is a "symbol".
Line: 22, 1st char: 1, "}" is a "symbol".
The symbol table contains:
Test1
add
a
b
c
b075020033@ubuntu:~/Desktop$ ./demo < Test2.java

Test2. java



Test3. java