

一些同學提出來的問題，希望可以幫到大家

(1)

該次作業僅僅要我們把輸入的檔案 印出到銀幕上或檔案中
而不必做出 **symbol table** 嗎？

像是能夠精確地判斷 ID 是否曾經有無被宣告過
或是每一個 **token** 所代表的意思
以便未來再做 **yacc** 時候會用到

答：

這次作業當然不僅僅只是要求你們將 輸入 的檔案印到螢幕上或檔案中
還需要你們將程式中的錯誤給抓出來
為了將錯誤給抓出來，你們當然必須做出 **Symbol Table** 來幫助你們找出錯誤的
所在
而你們這次的作業的標準是必須要能處理作業規定上的 7 項準則，通過 11 筆測
試檔的試驗，以及助教 **demo** 時所設計的測試檔

(2)

Lex 本身不會自己產生 **symbol table** 嗎??

答:

沒錯, **lex** 並不會自己產生 **symbol table**
而其實 **symbol table** 只是一個概念。實作方式可利用 **Array** 來完成
我幫你 google 了 "**lex symbol table**"
第一筆結果有範例
連結: <http://cs1.bradley.edu/public/jcm/cs518LexYacc.html>

裡面提到：

```
#define NSYMS 20 /* maximum number of symbols */
struct symtab
{ char *name;
  double value;
};
symtab symtable[NSYMS];
```

在這邊他是以陣列的方式來實作。

(3)

每當 match 到一個 token 時,裡面的動作有需要回傳任何值嗎?

ex: return(PROGRAM)

其中 PROGRAM 為自己事先定義(#define)好的常數

答：

回傳值是為了與 Yacc 做溝通的一種方式

本次作業不需要與 Yacc 做溝通，所以此部分是不需要的。

(4)

關於作業的說明檔(100-CompilerLab-SimplePascal-Scanner.pdf)中所提到的

4. 你/妳如何處理這份規格書上的問題

是要我們回答說明檔所提到的 7 種情形之下,是如何做處理的嗎?

答：

是的，請回答這 7 項準則你是如何做處理的。

當然囉，如果你還有在這次的作業上遇到其他的問題，也可以一並告訴助教你是如何處理的。

(5)

有關於解決錯誤的方法

是要我們設計一個所有可能錯誤的 pattern 來進行 match

之後再印出錯誤?

還是要將前面某一些字給刪除 之後再從新找尋 token??

如此一來後面就可能會出現正確的 token

ex: #abc 即是先把#給刪除,之後才重新 match 一遍

答：

我認為這個問題關係到你如何設計演算法

"設計一個所有可能錯誤的 pattern 來進行 match 之後再印出錯誤?" 可能是一個方法

但我無法向你保證這是否是最佳解或是最簡單的方法，以及這個方法的正確性同第一個問題，你只要能處理作業規定上的 7 項準則，通過 11 筆測試檔的試驗，以及助教 demo 時所設計的測試檔即算過關。

(5)

字串上的錯誤

在' '之內的字元若存在奇數個'

ex:'a'b' 中間有出現 1（奇數）個「'」

是否也是字串上的錯誤？？

同樣地,在註解上內成雙出現一對(* comment *)

ex:(* co(*mme*)nt *)

是否代表正確的註解？？

答：

根據作業上的規定， 'a'b' 是屬於錯誤的字串，因為兩個單引號之間的 單引號，並沒有使用跳脫字元 []

而另外 (* co(*mme*)nt *) 是屬於錯誤正確的註解，只要包覆在第一個 (* 與第一個出現的 *)中都視為註解

(6)

在規格書中的第二個範例:

Line3 3i : string ;

顯示了:

Line: 3, 1st char:, 3,"3i" is an invalid "ID".

範例中的錯誤修正的方法

若要依照規格書中第七個準則:哪些字元要被刪除

不就是"直接將批對到的錯誤給通通刪除"

(直接印出 invalid "ID",而跳到下一個尚未讀取字元的位置)

若是說刪除第一個字元(這裡是刪除"3"),再繼續做比對,則可能又顯示出:

Line: 3, 1st char:, 4,"i" is an "ID".

但似乎在這個範例中沒有出先這樣一個提示

我以上的解讀是否正確??

答：

我覺得你的解讀不是很正確

第七點準則指的是說，你可以告訴使用者該怎麼改正他們的程式

告訴使用者哪些字元要被刪除就是個範例

而不是自動幫忙修改程式，然後繼續比對顯示像 Line: 3, 1st char:, 4, "i" is an "ID". 的訊息

(6)

另外以下是測試檔案的部份：

test3.pas 的 line6 中：

如何判斷兩個數字之間的[+-]

是當運算子還是正負號

答：

這是作業的一部分喔，我不能告訴你怎麼判斷 XD

再問：

我是還是不出來要如何判斷一個 **statement** 中是當正負號還是當成是加減運算子

助教是否能給一些提示,或是更多的範例?或者是判斷的優先順序

答：

你有變數可以運用阿

如果你已經記住 +/- 之前沒有數字的話，那不就是當作正負號嗎

如果 +/- 的兩測是數字的話那就當作運算子囉

(我講的只是可能的兩個 **case**，不一定能符合所有的狀況。所以你還是要想一想)