

## 1. 版本

Lex, 版本 : 2.6.4-6

```
2.6.4-6 - debconf (16 (null)) de
c-compiler (0 (null)) libfl-dev
(0 (null)) flex:i386 (32 (null))
Provides:
2.6.4-6 - flex:i386 (= 2.6.4-6)
Reverse Provides:
flex:i386 2.6.4-6 (= 2.6.4-6)
libfl2,flex 2.5.39
libfl-dev,flex 2.5.39
libfl-dev,flex 2.6.4-6
```

Yacc 版本 : 3.0.4

```
Dependencies:
2:3.0.4.dfsg-1build1 - m4 (0 (null)) libc6 (2 2.15) libbison
Provides:
2:3.0.4.dfsg-1build1 - bison:i386 (= 2:3.0.4.dfsg-1build1)
Reverse Provides:
bison:i386 2:3.0.4.dfsg-1build1 (= 2:3.0.4.dfsg-1build1)
```

## 2. 作業平台

Oracle VM VirtualBox

Unbuntu

```
operatingsystem@operatingsystem-VirtualBox:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.4 LTS
Release:        18.04
Codename:       bionic
```

## 3. 執行方式

使用 makefile 進行編譯後，再用 ./a.out < testfile.pas 去執行不同的 test case 檔案。

## 4. 你/妳如何處理這份規格書上的問題

(1) 首先用 lex 切 token，包括 digit, unsigned, signed digit 和 reserve word 的字、空白、整數、小數、指數等等的 token.

(2) comment 的處理方式，用(\*\*)去判斷和將括號放進 symbol table 作資料結構的處理，主要是利用讀到\*)了沒去判斷結束以及有沒有讀到(\*來作判斷。

(3) 抓 reserve word 時忽略大小寫。

(4) 抓邏輯判斷、標點符號的 token.

(5) 用 function 將行數紀錄與增加、放 symbol table 等等

(6) 而後換到 yacc 的撰寫，首先是利用 extern 和 %union 來連接 lex 和 yacc 的變數，另外在宣告變數時，有另外加上需要顯示的 error message，而正常的 yytext, yyleng, yyerror 都有加入到變數宣告中。

(7) 首先是資料型態上的宣告，除了剛剛在 lex 中宣告的變數外，也需要額外的一些變數。

(8) 在處理 program 時，自然是將可能的 program name 標示進去，另外再加上；，因為沒有；就會顯示 error。

(9) 再來就是變數宣告時的文法了，i:, i:= 的判斷，用 semicolon 和 eq semicolon 去判斷。

(10) 其他就是將 id, 和 begin~end, array[1..8]的判斷加入。

(11) 在 main()中利用 yyparse()啟動。

(12) yyerror()讓 syntax error 的部分可以不要停滯並印出 error message.

## 5. 你/妳寫這個作業所遇到的問題

(1) Pascal 文法不太懂，摸滿久才弄出來，做的時候邊查邊理解。

(2) 因為文法不太懂，所以一開始連助教給的 error testfile 也不知道是錯在哪，寫的時候以為就錯那幾個，結果想不到有些也不合文法。

(3) lex 打的時候算滿順利的，但是在打 yacc 的時候就一直出現問題，像是 shift-shift conflict 和 shift-reduce conflict 的警告，好不容易把那個 warning 弄掉，結果顯示出來的結果不是會在奇怪的地方斷掉，就是跑到奇怪的判斷去，明明要他跑這行判斷，但是跑到不知到哪裡去了，真的很頭痛!!

(4) function 問題和 error message 顯示的問題，我有做豐富的 error message 顯示，但是不知道為甚麼當我在寫 yacc 的時候，有些 error message 會一直只跑某個 error message，然後改掉的話又會造成程式在遇到某些 error 時自動跳開，就是停止跑了!! 盡量修了，但是豐富的 error message 就不豐富了!!

(5) comment 處理：一直會顯示內文出來，結果最後雖然弄到不見了，也就是 line 會繼續 count 但是 yytext 會忽略他，但是還是會有大空白顯示出來，真的不知道為甚麼會這樣!!

## 6. Bonus

(1) 利用 Symbol table 作資料結構的處理，判斷 quote 的成對性：加入 structure 結構、lookup(), dump(), add(), print()等等的函式。

(2) If then 判斷：有作 if then 判斷，但是在判斷 else 的時候原本預計跑的 error message 怎麼也跑不出來，但是有成功跑過一次，是 Expect 'then' before else 的字眼，就是 if then else 有吃進去，並成功判斷 else 前面少了 then. 但是忘記截圖了....

7. 所有測試檔執行出來的結果,存成圖片或文字檔

```
m108-2@m108-2:~/Desktop/compiler/B051010001_hw2$ ./a.out < correct.pas
Line 1: program test;
Line 2: var
Line 3:
Line 4:   i, j: integer;
Line 5:   ans: array[0 .. 81] of integer;
Line 6: begin
Line 7:     i := -1 + 3;
Line 8:     j := +7*8;
Line 9:     ans[0] := 7;
Line 10:
Line 13:
Line 14:     for i:=1 to 9 do
Line 15:     begin
Line 16:         for j:=1 to i do
Line 17:             ans[i*9+j] := i*j;
Line 18:         end;
Line 19:
Line 20:     for i:=1 to 9 do
Line 21:     begin
Line 22:         for j:=1 to i do
Line 23:             if ( ans[i*9+j] mod 2 = 0) then
Line 24,1st char: 26, here 'syntax error'
Line 24:                 write(i, '*', j, '=', ans[i*9+j], ' '
m108-2@m108-2:~/Desktop/compiler/B051010001_hw2$ ./a.out < error1.pas
Line 1: program test;
Line 2: var
Line 3:   i: integer;
Line 4: begin
Line 5,1st char: 5, here 'syntax error'
Line 5,1st char: 8, here 'Need ':=' not ':' or '=' '
Line 5:   i = 3;
Line 6,1st char: 5, here 'syntax error'
Line 6,1st char: 8, here 'Need ':=' not ':' or '=' '
Line 6:   j = 4;
Line 7:   if (i > j)
Line 8,1st char: 15, here 'lose then'
Line 8:     Write('ok');
Line 9: end.
```

```
m108-2@m108-2:~/Desktop/compiler/B051010001_hw2$ ./a.out < error2.pas
Line 1: program test;
Line 2: var
Line 3:   i, j : integer;
Line 4: begin
Line 5:   i := 5*2;
Line 6:   j := 9;
Line 7:   if (i > j)
Line 8,1st char: 15, here 'lose then'
Line 8:     Write('ok');
Line 9: end.
m108-2@m108-2:~/Desktop/compiler/B051010001_hw2$ ./a.out < error3.pas
Line 1: program test;
Line 2: var
Line 3,1st char: 8, here 'syntax error'
Line 3,1st char: 11, here 'Need ':' not this '
Line 3:   i, j := integer;
Line 4: begin
Line 5:   i := 5;
Line 6: end
Line 7,1st char: 0, here 'lack . after end'
Line 7:
m108-2@m108-2:~/Desktop/compiler/B051010001_hw2$ ./a.out < error4.pas
Line 1: program test;
Line 2: var
Line 3:   i, j : integer;
Line 4:   c : string;
Line 5: begin
Line 6:   i := 5;
Line 7,1st char: 8, here 'syntax error'
Line 7:   c := 'aa';
Line 8,1st char: 8, here 'simpexp plus/minus term'
Line 8,1st char: 10, here 'Need ':=' not ':' or '=' '
Line 8:   i = i+c;
Line 9: end.
```