



在這個作業中，你（妳）必須完成一個符合【表一】文法的 **parser**。你（妳）可以修改第一個作業 — **scanner** 來切 **token**，丟給 **parser** 當作輸入。

你（妳）必須考慮下列這些問題：

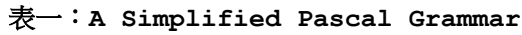
1. 你（妳）的 **parser** 在遇到 **error** 時，要能產生出好的 **error messages**，例如：發生 **error** 的行號、字元的位置和解釋 **error** 發生的原因。
2. 當 **parser** 遇到 **error** 時，要盡可能的處理完輸入，也就是說 **parser** 要遇到 **error** 要做 **recovery**。

你必須繳交下列檔案：

- ✧ 修改後的 **Scanner**，檔名為 — 你/妳的學號.**I**
- ✧ 你/妳的 **Parser**，檔名為 — 你/妳的學號.**y** (裡面需有註解，解釋如何處理 **statements**。)
- ✧ 你/妳的測試檔
- ✧ 所有的 **.c** 和 **.h** 檔 **makefile** 檔
- ✧ 一個 **Readme** 檔，裡面包含
 - **Lex, Yacc** 版本
 - 作業平台
 - 執行方式
 - 你/妳如何處理這份規格書上的問題
 - 你/妳寫這個作業所遇到的問題
 - 所有測試檔執行出來的結果，存成圖片或文字檔

請用壓縮軟體將上述這些檔案壓縮成一個檔案，檔名為 — 你/妳的學號，寄給助教

信件主旨請寫：[Compiler]Parser of 你/妳的學號

100-CompilerLab-SimplePascal-Parser (1)- 2/4



例子：

輸入	輸出
program test;	Line 1: program test;
var	Line 2: var
i : integer;	Line 3: i : integer;
begin	Line 4: begin
read(i);	Line 5: read(i);
end.	Line 6: end.

輸入	輸出
program test;	Line 1: program test;
var	Line 2: var
a: array [1 .. 5] of integer;	Line 3: a: array [1 .. 5] of integer;
b, c, d: integer;	Line 4: b, c, d: integer;
begin	Line 5: begin
b := 1;	Line 6: b := 1;
c := 3;	Line 7: c := 3;
a[2] :=c;	Line 8: a[2] :=c;
for b := 1 TO b*2+1 do	Line 9: for b := 1 TO b*2+1 do
BEGIN	Line 10: BEGIN
if (a[b] >= 2) then	Line 11: if (a[b] >= 2) then
WRITE(b);	Line 12: WRITE(b);
for d:= 1 To (1+2)*3 do	Line 13: for d:= 1 To (1+2)*3 do
WRITE(c);	Line 14: WRITE(c);
END;	Line 15: END;
end.	Line 16: end.

輸入	輸出
program test;	Line 1: program test;
var	Line 2: var
i, j := integer;	Line 3: i, j
begin	Line 3, 1st char: 8, syntax error at ":=".
i := 5;	Expect ':' not ':='
end.	Line 4: begin
	Line 5: i := 5;
	Line 5: end.

