



在這個作業中，你（妳）必須完成一個符合 `pascal` 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。

下方[表一]提供了簡易的 `pascal` 文法結構，供大家參考。此外，助教也提供 `pascal` 的完整文法結構給大家參考 (<https://condor.depaul.edu/ichu/csc447/notes/wk2/pascal.html>)。

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、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`。)
- ✧ 所有的 `.c` 和 `.h` 檔 `makefile` 檔
- ✧ 一個 `Readme` 檔(PDF)，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所提到可能會出現的 `error`
 - 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_`hw2`

2024(HW2)simple_pascal_parser.doc

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
(* one line comment *)	Line 4: i, j: integer;
i, j: integer;	Line 5: ans: array[0 .. 81] of integer;
ans: array[0 .. 81] of integer;	Line 6: begin
begin	Line 7: i := -1+3;
i := -1+3;	Line 8: j := +7*8;
j := +7*8;	Line 9: ans[0] := 7;
ans[0] := 7;	Line 14: for i:=1 to 9 do
(*	Line 15: begin
multiple lines comments	Line 16: for j:=1 to i do
do not show comments	Line 17: ans[i*9+j] := i*j;
*)	Line 18: end;
for i:=1 to 9 do	Line 20: for i:=1 to 9 do
begin	Line 21: begin
for j:=1 to i do	Line 22: for j:=1 to i do
ans[i*9+j] := i*j;	Line 23: if (ans[i*9+j] mod 2 = 0) then
end;	Line 24: write(i, '*', j, '=', ans[i*9+j]
for i:=1 to 9 do	, ' ');
begin	Line 25: writeln;
for j:=1 to i do	Line 26: end;
if (ans[i*9+j] mod 2 = 0) then	Line 27dd: end.
write(i, '*', j, '=', ans[i*9+j], '	
');	
writeln;	
end;	
end.	



輸入	輸出
program test;	Line 1: program test;
var	Line 2: var
i, j := integer;	Line 3, at char 8, ":" expected but "!=" found
begin	Line 4: begin
i := 5;	Line 5: i := 5;
end.	Line 5: end.

助教測資可能會出現的 error

1. 變數(或函數)未定義就使用，或是賦值型態與宣告型態不同(宣告 integer，賦值 string)
2. 結構有缺失，如遺漏左括號、then 前面沒有 if 等等
3. 缺少必要的符號(分號、句號等等)或使用錯誤的符號
4. 不同類別的變數相加，如：int + string

Error message 的呈現

關於偵測到 error 之後要呈現的 error message，要能解釋發生 error 的原因(如: 'a' is not defined)。若有不確定之處可參考 Online Pascal Compiler 的方式。

Online Pascal Compiler:

https://www.onlinegdb.com/online_pascal_compiler

作業的配分

- correct.pas 40%
- Error1 類型的錯誤 20%
- Error2~Error4 類型的錯誤 20%
- Readme 10%
- Demo 時的問答 10%
- Bonus:另外做出其他如 if...else 判斷