

Compiler hw2 (yacc) Readme B103040009 尹信淳

■ Yacc 版本:

```
bison (GNU Bison) 3.8.2
Written by Robert Corbett and Richard Stallman.

Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

■ Lex 版本:

```
flex 2.6.4
```

■ 作業平台:

```
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.4 LTS
Release:        22.04
Codename:       jammy
```

■ 執行方式:

1. make all 產生 a.out
2. ./a.out < test.pas

■ 規格書上所提到可能會出現的 error 的處理方式:

1. 遇到 error 時，要能產生出 error messages:
我用 yacc 提供的 error token，搭配 yyerror()，將 yacc 產生的 error message 印出來。
2. 遇到 error 時，除了要能產生出 error messages，還要能夠 recovery:
我用 yacc 提供的 error token 及 yyerrok，例如:
error TOKEN {yyerrok;}
讓 yacc 能在遇到 error 時跳到某個 token 後，換行繼續 parse.
3. 變數未定義就使用的問題:
我用一個 table 記錄定義過的 identifier，後續如有出現 table 沒有的 identifier 就會輸出 error msg.
4. 賦值型態與宣告型態不同的問題:

我有第二個 table 記錄 identifier 的型態，如果賦值型態不同，就會輸出 error msg.

5. 結構缺失:

在 .y 檔中定義正確的 grammar

6. 缺少必要的符號:

同 5.

7. 不同型態的變數相加:

類似 4.，如果兩變數型態不同，輸出 error msg.

■ 寫這個作業所遇到的問題:

我認為把這個作業寫好，需要對 pascal 語法及 yacc 的使用方式有足夠的認識，而我沒有讀完相關的文件就開始寫作業，我覺得有點事倍功半。很多問題是遇到之後，才去看文件、上網查，有時解決的不錯；有時會發現其實原本有其他更好的寫法或發現已經沒辦法在少量幅度的修改內完全修正錯誤。舉例來說，error token 的使用、grammar 的寫法、token 宣告等等都是。下次應該勤勞一點，多預留一些時間，做好準備再開始寫作業。

■ 測試檔執行結果:

1. correct.pas

```
michael@michael-virtual-machine:~/Desktop/B103040009_hw2$ ./a.out < ./testfile/correct.pas
Line 1: program test;
Line 2: var
Line 3: A comment starts
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;
Line10:     A comment starts
Line14:     for i:=1 to 9 do
Line15:       begin
Line16:         for j:=1 to i do
Line17:           ans[i*9+j] := i*j;
Line18:         end;
Line19:
Line20:     for i:=1 to 9 do
Line21:       begin
Line22:         for j:=1 to i do
Line23:           if ( ans[i*9+j] mod 2 = 0) then
Line24:             write(i, '*', j, '=', ans[i*9+j], ' ');
Line25:           writeln;
Line26:         end;
Line27: end.
```

2. error1.pas

```
michael@michael-virtual-machine:~/Desktop/B103040009_hw2$ ./a.out < ./testfile/error1.pas
Line 1: program test;
Line 2: var
Line 3:   i: integer;
Line 4: begin
Line 5: 1st char: 5, syntax error, unexpected '=', expecting ASSIGN
Line 5:   i = 3;
Line 6: 1st char: 5, syntax error, unexpected '=', expecting ASSIGN
Line 6:   j = 4;          syntax error: j undefined.
Line 7:   if (i > j) then      syntax error: j undefined.
Line 8:     Write('ok');
Line 9: end.
```

3. error2.pas

```
michael@michael-virtual-machine:~/Desktop/B103040009_hw2$ ./a.out < ./testfile/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) then
Line 8:     Write('ok');
Line 9: end.
```

4. error3.pas

```
michael@michael-virtual-machine:~/Desktop/B103040009_hw2$ ./a.out < ./testfile/error3.pas
Line 1: program test;
Line 2: var
Line 3: 1st char: 9, syntax error, unexpected ASSIGN, expecting ':' or ','
Line 3:   i, j := integer;
Line 4: begin
Line 5:   i := 5;
Line 6: 1st char: 4, syntax error, unexpected end of file, expecting '.'
Line 6: end
```

5. error4.pas

```
michael@michael-virtual-machine:~/Desktop/B103040009_hw2$ ./a.out < ./testfile/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:   c := 'aa';
Line 8: 1st char: 5, syntax error, unexpected '=', expecting ASSIGN
Line 8:   i = i+c;
Line 9: end.
```