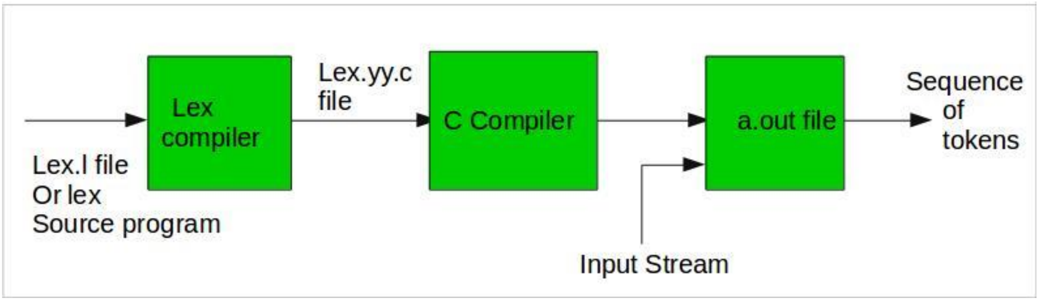
HW1 Simple Pascal Scanner using Lex
B08304029 邱品諺

1. Flex 2.6.4
2. Ubuntu 64-bit 20.04.4
3.



利用 lex 進行編譯的過程如上圖所示，首先先將 lex 檔編譯成 C 檔，再將 C 檔編譯成執行檔並引入輸入檔以 scan 輸入檔的 token。撰寫 lex 的方式主要要先定義 token 可能出現的種類，如 string、comment 等等，並在定義的部分規定其 regular expression，再來輸入流的 token 會由上到下依序對應定義的 token 種類，並執行其定義的 lex rule 部分，以印出各 token 所對應的種類及其合法或非法。

4.

(1) Reserved word

由於 Pascal 是 case-insensitive 的語言，所以一個 word 的每個字元有可能會是大寫或小寫，因此在 regular expression 的表示上需要利用[]來表示每個字元的可能，例如：and 需要以[Aa][Nn][Dd]來表示其 regular expression。

(2) Identifiers

合法的 identifiers 第一個字只能是英文字母或是底線符號，之後可以是英文字母、數字或底線符號。需要注意的是可能會將分號、逗號、冒號和括號等符號與前面的 identifier 合在一起表示，因此在 regular expression 的表示上需要特別處理。而除了第一個字不是英文字母或底線符號所組成的 identifier 為非法的以外，只要其中任何一個字元只要出現非英文字母、數字、底線符號即為非法 identifiers。因此在 regular expression 的表示上無法僅利用第一個字不為英文字母及底線符號的線索判斷，也需要考慮後續所出現的可能字元。另外，由於規定長度不可超過 15，而只需在合法 identifier 所定義的規則（lex rule）下，利用 yyleng 判斷長度是否超過 15 即可。

(3) Symbols

利用 regular expression 表示可能出現的符號，如果出現不只 1 byte 的符號利用 |（alternative）分開即可。需特別注意的是逗號（,）不在列表中，所以本程式特別寫了一個定義 comma，使得在 scan 時可將其忽略。

(4) 實數

實數除了要考慮有無正負外，還要考慮有無小數及有無 exponential 的部分，且小數不可多個 0 結尾，如：1.00，另外，整數除了 0 以外，其餘不可以 0 開頭，如 003，而這些利用 regular expression 即可完整表示。另外，非法的實數其實在 regular expression 的表示上與合法的類似，只要不加以限制 0 出現在非法的位置，如上述例子，因此利用 regular expression 即可完整表示。

(5) Quoted String

由於只要在引號（″）內的任何字元皆為字串，因此在 regular expression 的表示上非常單純，但是需要考慮一些特別情形，例如：空字串、字串中有空白、跳脫字元等等。跳脫字元的處理上，欲將’You’’ll see’顯示為’You’ll see’，因此需要在 lex rule 的部分特別處理，本程式中使用迴圈跑過欲顯示的字串，當遇到引號（'）時便向後檢查是否也是引號（'），如果是的話便將引號後的所有字源都向前挪一格，且在最後一個被挪動的加上字串結尾（'\0），挪動完畢後便在向後檢查是否仍有跳脫字元直到結束。而在非法字串的表示上需要考慮引號未刮完全，或是字串中只有奇數個引號的部分，例如：’ab、’a’b’, ab’等等。另外，長度 30 以上的字串被視為非法字串，只需利用 yyleng 判斷 token 長度即可。

(6) 註解

只要在(*及*)內，除非中間出現*)，否則皆視為合法註解，因此如：(* co(*mme*)nt *)、 (*ab*)**)皆為非法的註解，而(* co(*ma*)為合法註解。標示註解合法或非法上，本程式優先處理非法的註解，剩下的即為合法註解。另外，由於註解可以跨行，因此除了在 regular expression 的表示上需注意外，在行數的表示上需要調整。

(7) 數學運算式中實數及符號的分離

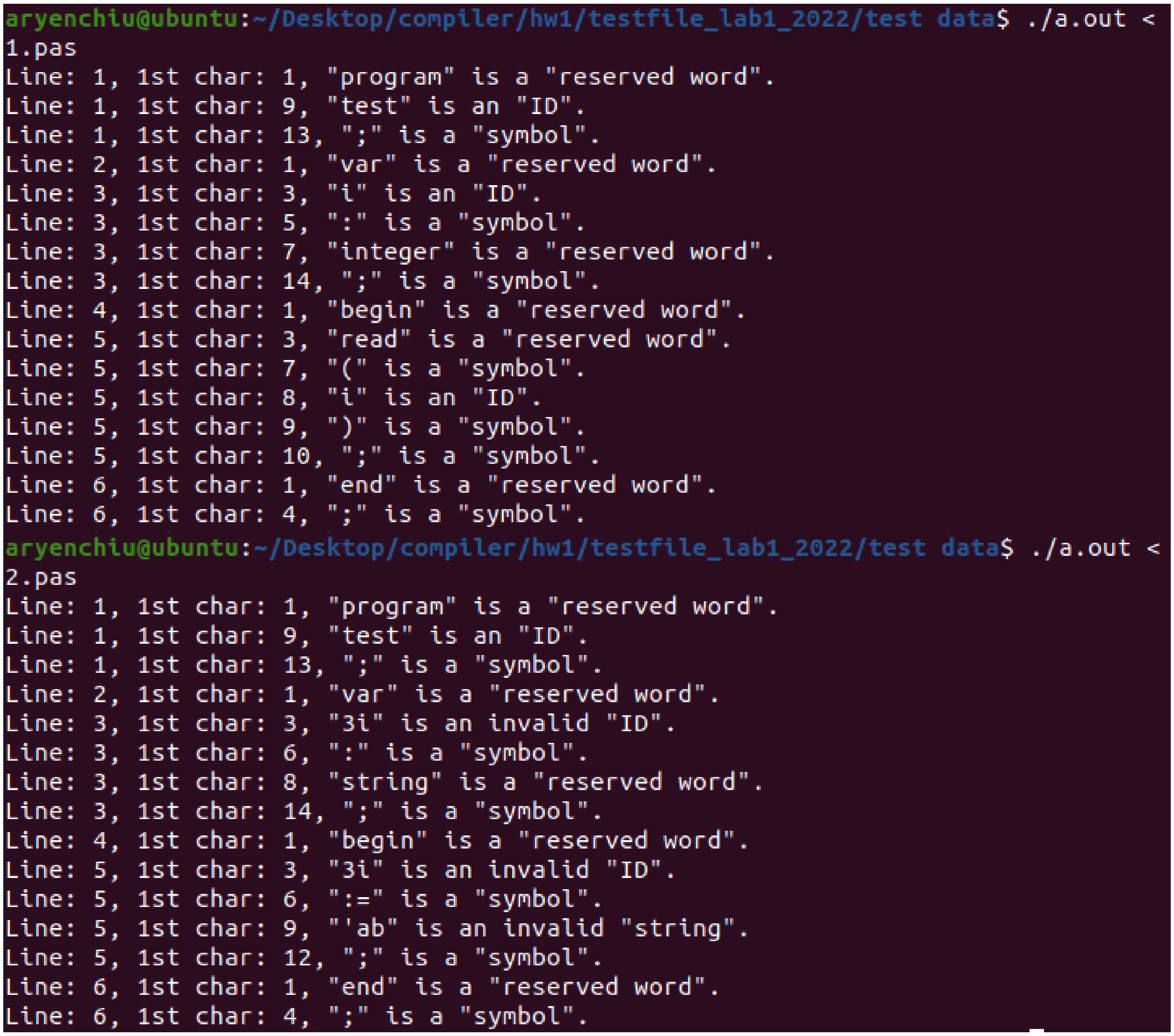
由於在賦值時可能給予變數一串數學運算式，且實數有正負，如 i = 1+2。因此需特別分離何者為實數何者為運算符號，本程式設計將所有數學運算式先判斷為非法的 identifier，再從中判斷如果所有字元皆為數字及符號的話即為數學運算式。首先，如果正負符號前面如果是數字的話，即表示其為運算符號而非表示實數正負的符號，而乘號及除號絕對為運算符號，另外，判斷實數的部分需先考慮其數字前符號的前一個位置是否為數字，如果為數字的話即表示其為無正負的實數，反之即為有正負的實數，此部分本程式利用一個字串及迴圈將該實數取出並顯示。

(8) Scanner 如何在發生錯誤時做 recover？

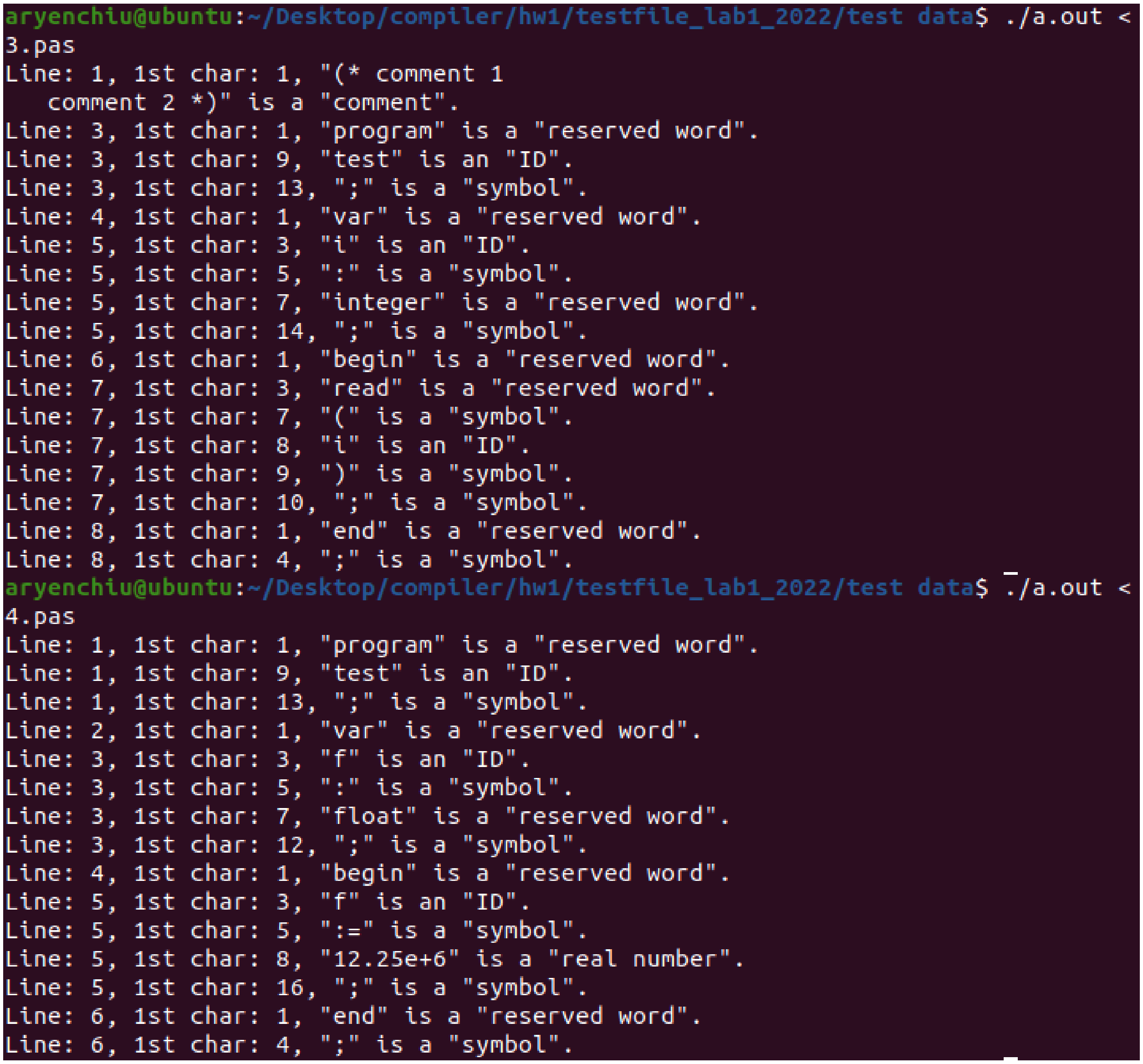
首先需要先能夠偵測出錯誤，本程式針對每個 token 的型態表示其合法及非法的 regular expression，因此可判斷出該 token 為何種型態且為合法或非法。所以當出現非法的 token 時，可持續辨識接下來的 token 直到結束，中途並不會暫停。

5. 實做一個 scanner 並不容易，首先，想出一個 regular expression 需要絞盡腦汁外，還有可能存在有些特例因而無法表示所有的情況，所以需要在 lex rule 的部分撰寫規則進行分類，或是分析、拆解字元等等。

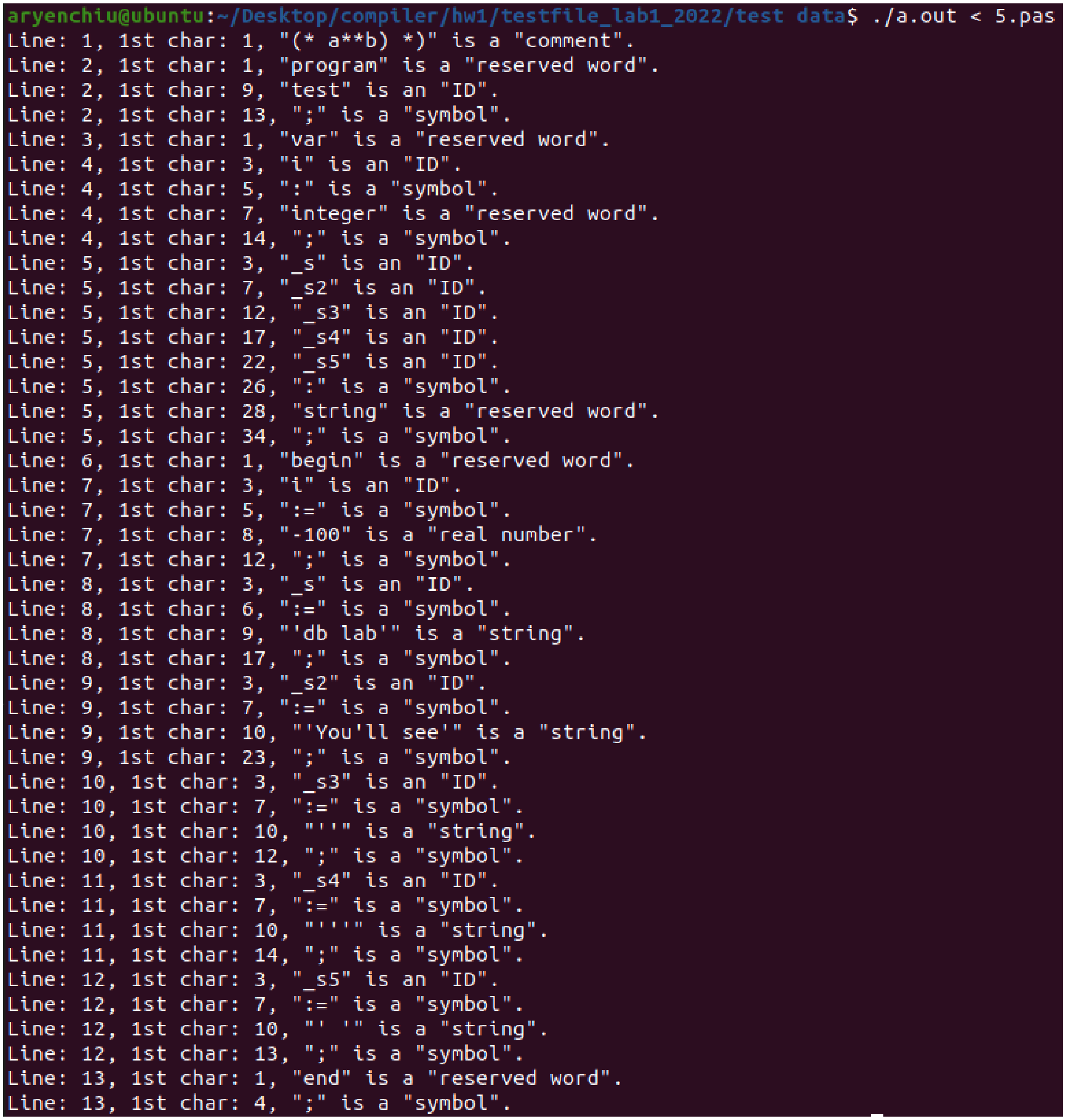
6.



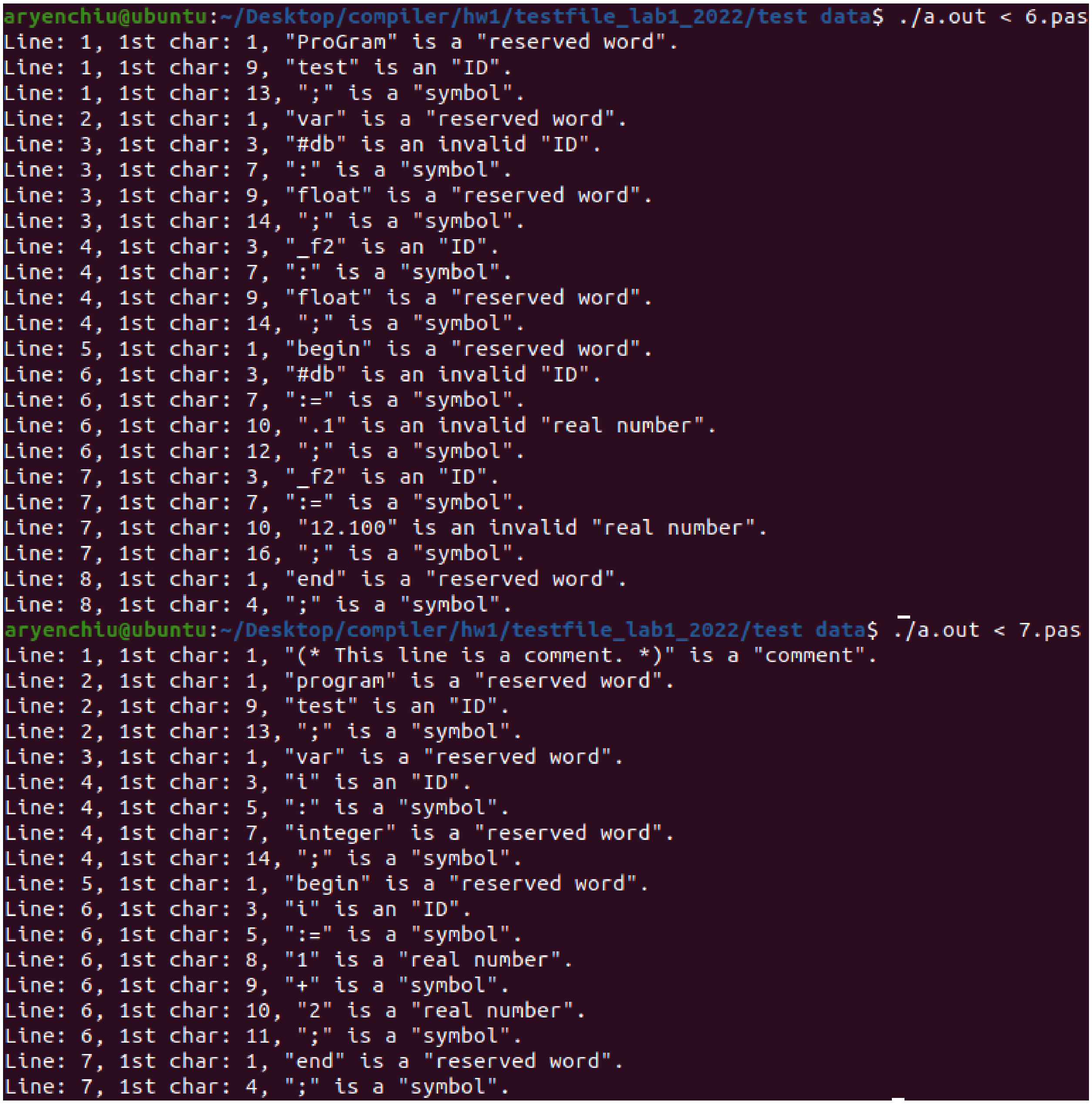
```
aryenchiu@ubuntu:~/Desktop/compiler/hw1/testfile_lab1_2022/test data$ ./a.out <
1.pas
Line: 1, 1st char: 1, "program" is a "reserved word".
Line: 1, 1st char: 9, "test" is an "ID".
Line: 1, 1st char: 13, ";" is a "symbol".
Line: 2, 1st char: 1, "var" is a "reserved word".
Line: 3, 1st char: 3, "i" is an "ID".
Line: 3, 1st char: 5, ":" is a "symbol".
Line: 3, 1st char: 7, "integer" is a "reserved word".
Line: 3, 1st char: 14, ";" is a "symbol".
Line: 4, 1st char: 1, "begin" is a "reserved word".
Line: 5, 1st char: 3, "read" is a "reserved word".
Line: 5, 1st char: 7, "(" is a "symbol".
Line: 5, 1st char: 8, "i" is an "ID".
Line: 5, 1st char: 9, ")" is a "symbol".
Line: 5, 1st char: 10, ";" is a "symbol".
Line: 6, 1st char: 1, "end" is a "reserved word".
Line: 6, 1st char: 4, ";" is a "symbol".
aryenchiu@ubuntu:~/Desktop/compiler/hw1/testfile_lab1_2022/test data$ ./a.out <
2.pas
Line: 1, 1st char: 1, "program" is a "reserved word".
Line: 1, 1st char: 9, "test" is an "ID".
Line: 1, 1st char: 13, ";" is a "symbol".
Line: 2, 1st char: 1, "var" is a "reserved word".
Line: 3, 1st char: 3, "3i" is an invalid "ID".
Line: 3, 1st char: 6, ":" is a "symbol".
Line: 3, 1st char: 8, "string" is a "reserved word".
Line: 3, 1st char: 14, ";" is a "symbol".
Line: 4, 1st char: 1, "begin" is a "reserved word".
Line: 5, 1st char: 3, "3i" is an invalid "ID".
Line: 5, 1st char: 6, ":=" is a "symbol".
Line: 5, 1st char: 9, "'ab" is an invalid "string".
Line: 5, 1st char: 12, ";" is a "symbol".
Line: 6, 1st char: 1, "end" is a "reserved word".
Line: 6, 1st char: 4, ";" is a "symbol".
```



```
aryenchiu@ubuntu:~/Desktop/compiler/hw1/testfile_lab1_2022/test data$ ./a.out <
3.pas
Line: 1, 1st char: 1, "(* comment 1
   comment 2 *)" is a "comment".
Line: 3, 1st char: 1, "program" is a "reserved word".
Line: 3, 1st char: 9, "test" is an "ID".
Line: 3, 1st char: 13, ";" is a "symbol".
Line: 4, 1st char: 1, "var" is a "reserved word".
Line: 5, 1st char: 3, "i" is an "ID".
Line: 5, 1st char: 5, ":" is a "symbol".
Line: 5, 1st char: 7, "integer" is a "reserved word".
Line: 5, 1st char: 14, ";" is a "symbol".
Line: 6, 1st char: 1, "begin" is a "reserved word".
Line: 7, 1st char: 3, "read" is a "reserved word".
Line: 7, 1st char: 7, "(" is a "symbol".
Line: 7, 1st char: 8, "i" is an "ID".
Line: 7, 1st char: 9, ")" is a "symbol".
Line: 7, 1st char: 10, ";" is a "symbol".
Line: 8, 1st char: 1, "end" is a "reserved word".
Line: 8, 1st char: 4, ";" is a "symbol".
aryenchiu@ubuntu:~/Desktop/compiler/hw1/testfile_lab1_2022/test data$ ./a.out <
4.pas
Line: 1, 1st char: 1, "program" is a "reserved word".
Line: 1, 1st char: 9, "test" is an "ID".
Line: 1, 1st char: 13, ";" is a "symbol".
Line: 2, 1st char: 1, "var" is a "reserved word".
Line: 3, 1st char: 3, "f" is an "ID".
Line: 3, 1st char: 5, ":" is a "symbol".
Line: 3, 1st char: 7, "float" is a "reserved word".
Line: 3, 1st char: 12, ";" is a "symbol".
Line: 4, 1st char: 1, "begin" is a "reserved word".
Line: 5, 1st char: 3, "f" is an "ID".
Line: 5, 1st char: 5, ":=" is a "symbol".
Line: 5, 1st char: 8, "12.25e+6" is a "real number".
Line: 5, 1st char: 16, ";" is a "symbol".
Line: 6, 1st char: 1, "end" is a "reserved word".
Line: 6, 1st char: 4, ";" is a "symbol".
```



```
aryenchiu@ubuntu:~/Desktop/compiler/hw1/testfile_lab1_2022/test data$ ./a.out < 5.pas
Line: 1, 1st char: 1, "(* a**b) *)" is a "comment".
Line: 2, 1st char: 1, "program" is a "reserved word".
Line: 2, 1st char: 9, "test" is an "ID".
Line: 2, 1st char: 13, ";" is a "symbol".
Line: 3, 1st char: 1, "var" is a "reserved word".
Line: 4, 1st char: 3, "i" is an "ID".
Line: 4, 1st char: 5, ":" is a "symbol".
Line: 4, 1st char: 7, "integer" is a "reserved word".
Line: 4, 1st char: 14, ";" is a "symbol".
Line: 5, 1st char: 3, "_s" is an "ID".
Line: 5, 1st char: 7, "_s2" is an "ID".
Line: 5, 1st char: 12, "_s3" is an "ID".
Line: 5, 1st char: 17, "_s4" is an "ID".
Line: 5, 1st char: 22, "_s5" is an "ID".
Line: 5, 1st char: 26, ":" is a "symbol".
Line: 5, 1st char: 28, "string" is a "reserved word".
Line: 5, 1st char: 34, ";" is a "symbol".
Line: 6, 1st char: 1, "begin" is a "reserved word".
Line: 7, 1st char: 3, "i" is an "ID".
Line: 7, 1st char: 5, ":=" is a "symbol".
Line: 7, 1st char: 8, "-100" is a "real number".
Line: 7, 1st char: 12, ";" is a "symbol".
Line: 8, 1st char: 3, "_s" is an "ID".
Line: 8, 1st char: 6, ":=" is a "symbol".
Line: 8, 1st char: 9, "'db lab'" is a "string".
Line: 8, 1st char: 17, ";" is a "symbol".
Line: 9, 1st char: 3, "_s2" is an "ID".
Line: 9, 1st char: 7, ":=" is a "symbol".
Line: 9, 1st char: 10, "'You'll see'" is a "string".
Line: 9, 1st char: 23, ";" is a "symbol".
Line: 10, 1st char: 3, "_s3" is an "ID".
Line: 10, 1st char: 7, ":=" is a "symbol".
Line: 10, 1st char: 10, "''" is a "string".
Line: 10, 1st char: 12, ";" is a "symbol".
Line: 11, 1st char: 3, "_s4" is an "ID".
Line: 11, 1st char: 7, ":=" is a "symbol".
Line: 11, 1st char: 10, "''''" is a "string".
Line: 11, 1st char: 14, ";" is a "symbol".
Line: 12, 1st char: 3, "_s5" is an "ID".
Line: 12, 1st char: 7, ":=" is a "symbol".
Line: 12, 1st char: 10, "' '" is a "string".
Line: 12, 1st char: 13, ";" is a "symbol".
Line: 13, 1st char: 1, "end" is a "reserved word".
Line: 13, 1st char: 4, ";" is a "symbol".
```



```
aryenchiu@ubuntu:~/Desktop/compiler/hw1/testfile_lab1_2022/test data$ ./a.out < 6.pas
Line: 1, 1st char: 1, "ProGram" is a "reserved word".
Line: 1, 1st char: 9, "test" is an "ID".
Line: 1, 1st char: 13, ";" is a "symbol".
Line: 2, 1st char: 1, "var" is a "reserved word".
Line: 3, 1st char: 3, "#db" is an invalid "ID".
Line: 3, 1st char: 7, ":" is a "symbol".
Line: 3, 1st char: 9, "float" is a "reserved word".
Line: 3, 1st char: 14, ";" is a "symbol".
Line: 4, 1st char: 3, "_f2" is an "ID".
Line: 4, 1st char: 7, ":" is a "symbol".
Line: 4, 1st char: 9, "float" is a "reserved word".
Line: 4, 1st char: 14, ";" is a "symbol".
Line: 5, 1st char: 1, "begin" is a "reserved word".
Line: 6, 1st char: 3, "#db" is an invalid "ID".
Line: 6, 1st char: 7, ":=" is a "symbol".
Line: 6, 1st char: 10, ".1" is an invalid "real number".
Line: 6, 1st char: 12, ";" is a "symbol".
Line: 7, 1st char: 3, "_f2" is an "ID".
Line: 7, 1st char: 7, ":=" is a "symbol".
Line: 7, 1st char: 10, "12.100" is an invalid "real number".
Line: 7, 1st char: 16, ";" is a "symbol".
Line: 8, 1st char: 1, "end" is a "reserved word".
Line: 8, 1st char: 4, ";" is a "symbol".
aryenchiu@ubuntu:~/Desktop/compiler/hw1/testfile_lab1_2022/test data$ ./a.out < 7.pas
Line: 1, 1st char: 1, "(* This line is a comment. *)" is a "comment".
Line: 2, 1st char: 1, "program" is a "reserved word".
Line: 2, 1st char: 9, "test" is an "ID".
Line: 2, 1st char: 13, ";" is a "symbol".
Line: 3, 1st char: 1, "var" is a "reserved word".
Line: 4, 1st char: 3, "i" is an "ID".
Line: 4, 1st char: 5, ":" is a "symbol".
Line: 4, 1st char: 7, "integer" is a "reserved word".
Line: 4, 1st char: 14, ";" is a "symbol".
Line: 5, 1st char: 1, "begin" is a "reserved word".
Line: 6, 1st char: 3, "i" is an "ID".
Line: 6, 1st char: 5, ":=" is a "symbol".
Line: 6, 1st char: 8, "1" is a "real number".
Line: 6, 1st char: 9, "+" is a "symbol".
Line: 6, 1st char: 10, "2" is a "real number".
Line: 6, 1st char: 11, ";" is a "symbol".
Line: 7, 1st char: 1, "end" is a "reserved word".
Line: 7, 1st char: 4, ";" is a "symbol".
```