**DESIGN AND IMPLEMENTATION OF COMPILER Homework 1**

B093040051 劉世文

1. **Lex 版本:**

flex 2.6.4

1. **作業平台**

Ubuntu 22.04 LTS

1. **執行方式**
2. 打開 terminal （終端機）。
3. 使用 **cd** 命令切換至包含 Lex 檔案的目錄。
4. 輸入指令 **make all** 來編譯 Lex 文件。
5. 執行編譯後的程式，並將待掃描的 Pascal 檔案作為輸入：**./a.out < file\_to\_be\_scanned.pas**。
6. 程式將輸出掃描結果。
7. **你/妳如何處理這份規格書上的問題**

* **保留字(Reserved words)**

首先明確列出了 Pascal 語言的所有保留字，每個保留字都使用了正則表達式來匹配，並且考慮到了 Pascal 的大小寫不敏感特性，例如保留字 **program**，將保留字例如 **program**的正則表達式被設計為 **[pP][rR][oO][gG][rR][aA][mM]**，這樣無論是大小寫都可以被正確匹配。在定義好這些保留字的正則表達式之後，使用統一的關鍵字 **reserved\_words** 來代表所有保留字，透過 **{absolute}|{and}|{begin}|**...，使用 **|**（OR）連接起來形成。讓 Lex 在掃描文本時，能一次偵測所有保留字，並在匹配到保留字時觸發相應的動作。在此次作業中，相應的動作是呼叫 **output\_result** 函數，將保留字逐一印出。

* **識別字(Identifiers)**

識別字的匹配採用了正則表達式 [a-zA-Z\_][a-zA-Z0-9\_]{0,14}，這意味著：

1. 第一個字元必須是字母（無論大小寫）或底線 \_。
2. 在第一個字元之後，識別字可以包含字母（無論大小寫）、數字和底線 \_。
3. 識別字的總長度限制為最多 15 個字元。

對於不合法的識別字（如以數字開頭的識別字），文件中定義了 **invalid\_identifiers** 正則表達式 [a-zA-Z0-9#]+ 來捕捉可能的錯誤。

* **符號(Symbols)**

符號被定義為 :=|==|<=|>=|[\;\:\(\)\[\]\+\-\\*\/\.\=\<\>]，針對單字元符號和雙字元符號分別處理，由於雙字元符號包含單字元符號本身，因此將雙字元符號列於單字元符號以前，以此區分並進行適當的處理。

* **實數(Real numbers)**

實數的正則表達式為[+-]?({digit}+(\.{digit}+))|({digit}+(\.{digit}+)?)([eE][+-]?{digit}+) 包含了幾個主要部分：

1. [+-]?：可選的正負號，表示實數可以是正數也可以是負數。
2. {digit}+(.{digit}+)：至少一位數字，跟著一個小數點和至少一位小數，匹配如 "1.0" 或 "123.456" 等格式的實數。
3. {digit}+(.{digit}+)?([eE][+-]?{digit}+)：匹配科學記數法表示的實數，如 "1.23e+10"、"7E-10" 等，允許實數後面跟有指數部分，指數部分由 'e' 或 'E' 開頭，可能包含正負號，後面跟著至少一位數字表示指數。

* **字串常數(Quoted strings)**

字串常數正則表達式為\'([^\']|\'\'){0,28}\'，包含的意義是**：**

1. \'：匹配開始和結束的單引號。
2. ([^\']|\'\')：匹配任何非單引號的字元。或是匹配兩個連續的單引號，代表字元串內的單引號字元。
3. {0,28}：重複前面的([^\']|\'\') 0 到 28 次，意味著支持的字元串長度最多為 30 個字元，包含開始和結束的單引號。

* **註解(Comments)**

註解的處理通過定義一個特定的開始條件 **COMMENT**，Lex 能在遇到註解開始標記 **(\*** 時切換到一個專門處理註解的狀態，直到遇到註解結束標記 **\*)**。

**"(\*" { BEGIN(COMMENT); }**

**<COMMENT>"\*)" { BEGIN(INITIAL); }**

當匹配到 **(\*** 時，**BEGIN(COMMENT)**; 表示進入 **COMMENT** 狀態，直到匹配到 **\*)** ，**BEGIN(INITIAL)**; 表示返回到初始狀態，退出註解處理狀態。

在 COMMENT 模式下，所有內容基本上都被忽略，直到遇到註解結束標記。然而，為了準確計算行數，遇到換行字元 \n 需要特別處理：

**<COMMENT>\n { lineCount++; }**

**<COMMENT>. { }**

每當在註解中遇到換行字元時，lineCount++ 用於計算文件的行數，以確保行數的準確性，並且使用 **.** 匹配，所有字元在註解中被忽略，不進行任何處理。

* **Error handling**

對於其他未明確定義的token，因而無法被任何規則匹配到的字元會統一由**Other** **.**捕捉，避免出現 Crash 的情況，並且在輸出中，確認錯誤發生的地方，以及是否需要刪除字元。

1. **你/妳寫這個作業所遇到的問題**

在 Lex 中處理大小寫不敏感的詞匹配時，需要為每個保留字定義正則表達式，這一過程較為繁瑣，而且格式較為混亂，後來改由個別定義保留字後，再由**{reserved\_words}** 統一呼叫對應的函式，以維持較簡潔和可讀的程式碼。

在匹配註解時，因為沒有stateful的方法，一直沒有辦法成功完成，後來透過定義**%x COMMENT**，在匹配到**(\***和**\*)**的時候，分別開始和結束註解狀態，以此完成註解的功能。後來因為註解的處理會忽略行數，所以也需要在註解的狀態特別處理。

在處理像 1.0+2.0 這樣的表達式時，需要將它們分別識別為兩個實數和一個加號符號。若以原本的 **{real\_num}** 及 **{symbol}** 匹配，會得到1.0為實數和+2.0為實數的結果，因此在應對這個問題，需要額外定義實數連著符號的規則 **{real\_num\_sym}** 以匹配實數和符號，先捕捉1.0+，並且拆解為1.0為實數和+為符號，接續執行lex將2.0匹配為實數，就可以避免這個問題的發生。

因為匹配規則的順序導致錯誤匹配，由於原先的 **{invalid\_identifiers}** 的優先度高於 **{invalid\_real\_num}** ，因此在某些情況下，不合法的實數會被匹配為不合法的識別字，同樣的問題也發生在其他規則當中，再重新調整和測試後，就解決了這個問題。

1. **所有測試檔執行出來的結果，存成圖片或文字檔**

* 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".

* 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 "quoted 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".

* 3.pas

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".

* 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".

* 5.pas

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: 6, "\_s2" is an "ID".

Line: 5, 1st char: 10, "\_s3" is an "ID".

Line: 5, 1st char: 14, "\_s4" is an "ID".

Line: 5, 1st char: 18, "\_s5" is an "ID".

Line: 5, 1st char: 22, ":" is a "symbol".

Line: 5, 1st char: 24, "string" is a "reserved word".

Line: 5, 1st char: 30, ";" 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 an invalid "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 "quoted 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 "quoted 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 "quoted 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 "quoted 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 "quoted 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".

* 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 a "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".

* 7.pas

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 an invalid "real number".

Line: 6, 1st char: 9, "+" is a "symbol".

Line: 6, 1st char: 10, "2" is an invalid "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".