

系統程式-作業一

Lexical Analysis 實作

110 學年度第一學期

老師：賴建宏

學號：10827216

班級：資訊三乙

姓名：王立翔

壹、開發環境

作業系統：Window 10

程式語言：C++11

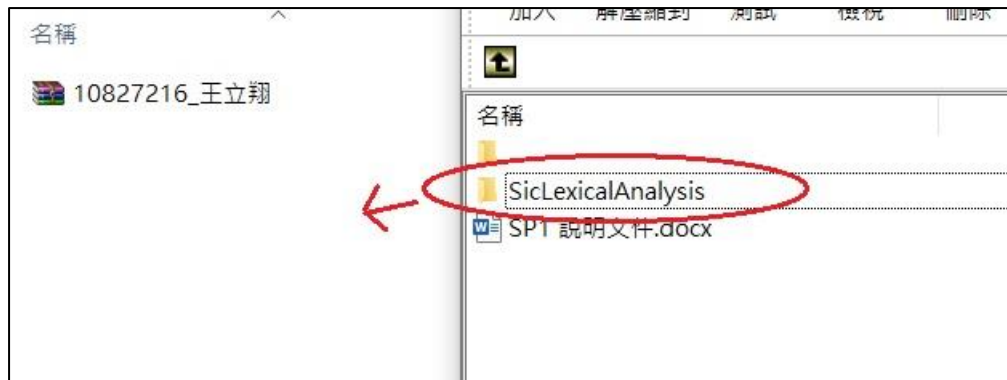
編譯器：MinGW

IDE：Code::Blocks 17.12

實作內容：**SIC** Lexical Analysis

貳、執行流程

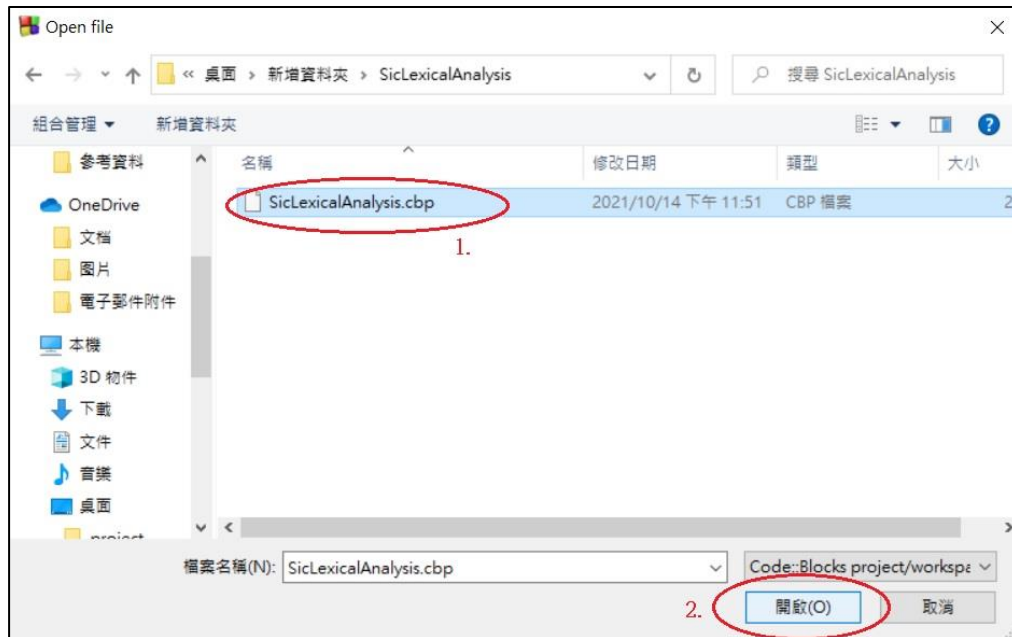
1. 將壓縮檔內 **SicLexicalAnalysis** 資料夾解壓縮



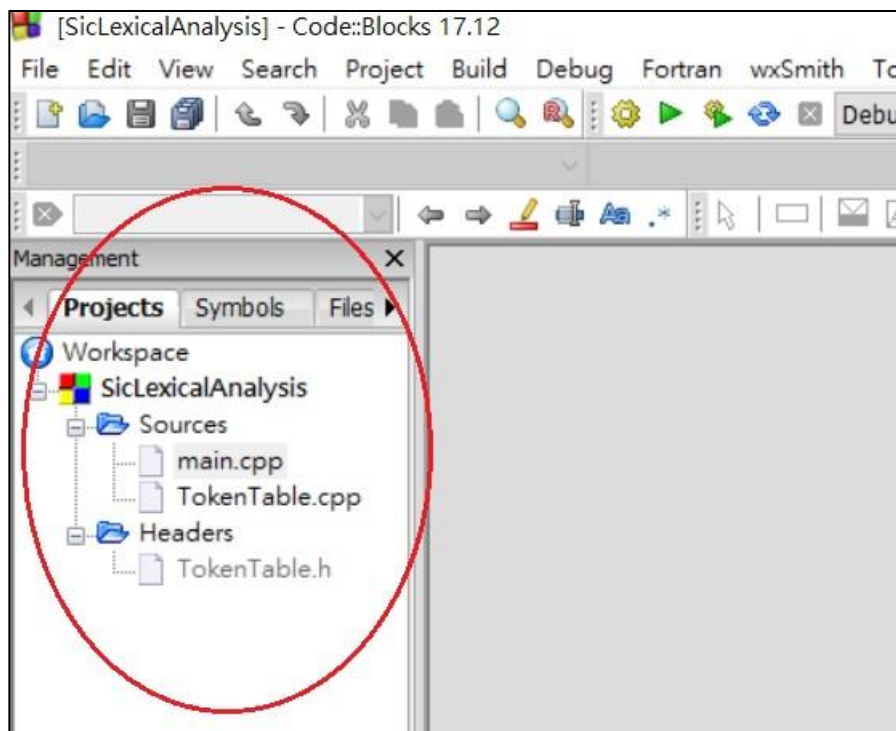
2. 開啟 Code Blocks，點選畫面正中央 **Open an existing project**



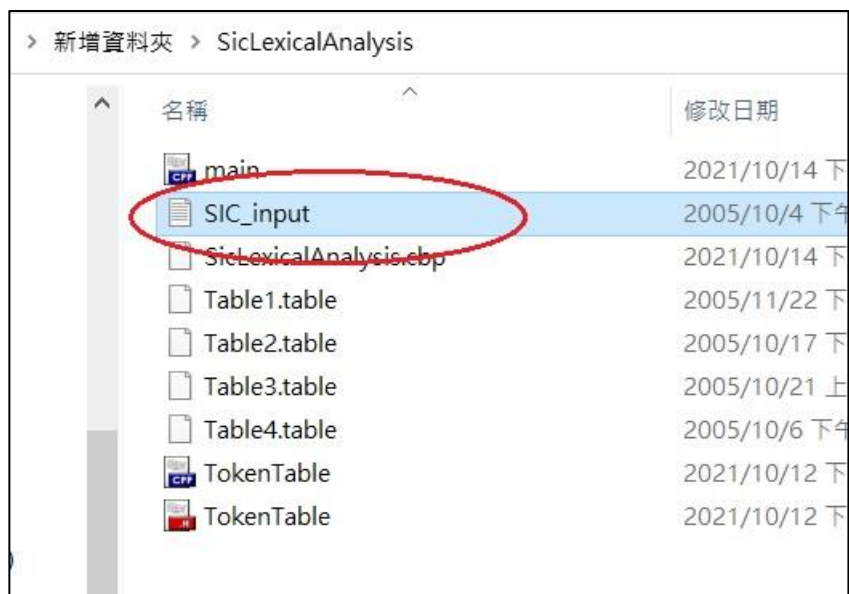
3. 找到剛剛解壓縮的資料夾路徑，選取其內的 **SicLexicalAnalysis.cbp** 檔案，並按下開啟



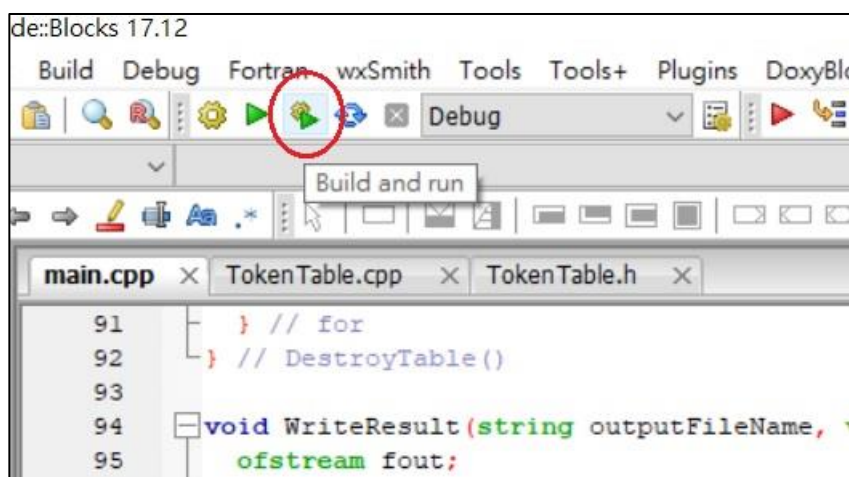
4. 展開左方工作區即可看到 project 內 **.h/.cpp** 檔案



5. 將欲讀取的 input 檔放進 **SicLexicalAnalysis** 資料夾內（與 main.cpp、SicLexicalAnalysis.cbp 檔案同一路徑）



6. 回到 Code Blocks，按下上方 **Build and run** 以執行程式



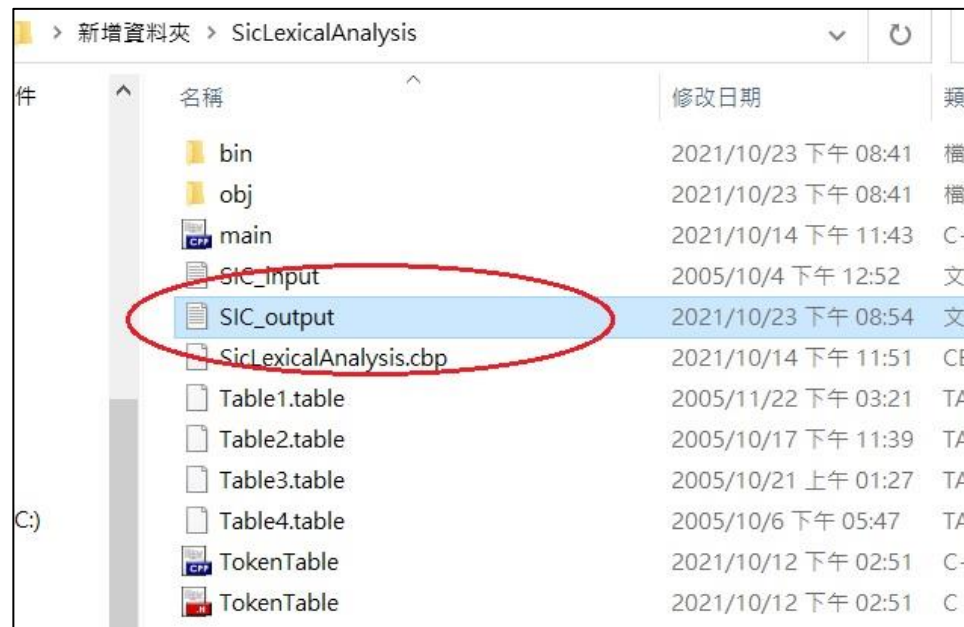
7. 於執行畫面輸入欲讀入檔案的檔名後，按下 Enter



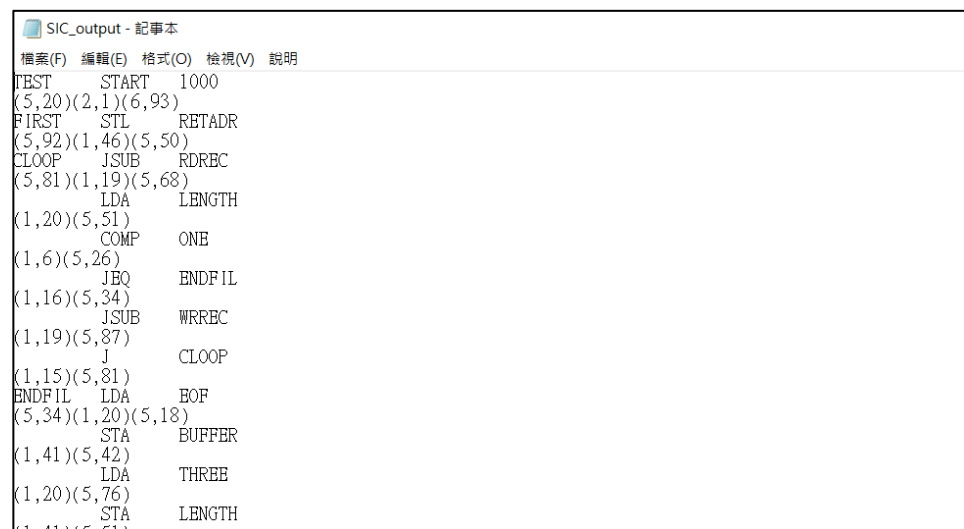
8. 畫面顯示分析完畢提示字後，輸入欲輸出檔案之檔名後，按下 Enter

```
"C:\Users\paul wang\Desktop\新增資料夾\SicLexicalAnalysis\bin\Debug\SicLexica
Input a source name: SIC_input.txt
Analysis completely!
Input output file name: SIC_output.txt
```

9. 回到 SicLexicalAnalysis 資料夾，即可發現程式生成一個新的 output 檔案



10. 以記事本打開確認輸出結果



參、程式設計

一、功能

輸入：SIC 組合語言程式碼

輸出：原始碼切完 Token 後的結果，輸出每個 Token 其所在 Table 編號以及其於 Table 內的位置

讀入並分析原始碼，將讀到字串切割成最小單位 Token，並將 Token 分成七類，分別是指令(Instruction)、假指令(Pseudo)、暫存器(Register)、分隔符(Delimiter)、符號(Symbol)、數字(Integer/Real)、字串(String)，並分別以一個 Table 儲存同類 Token，編號由以上類型順序編號 1~7，其中編號 1~4 的 Table 為普通陣列容器，index 由 1 開始存放，執行過程中不可變動；編號 5~7 的 Table 為 Hash Table，容量限制為 100，index 由 0 開始，程式執行過程中可以插入。

二、流程

1. 從原始碼讀取字元，加入至 TKBUF，直到找到 White-space、EOF 或分隔符號。

2-1. 若找到 White-space 或 EOF，檢查 TKBUF 大小是否大於 0，是則將 TKBUF 內容存入至 Table，並將 Token 資訊記錄下來，存至 LBUF；承上步，若找到字元是換行或 EOF，則檢查 LBUF 大小是否大於 0，是則輸出 LBUF 內容（一行原始碼的 Tokens 資訊）。

2-2. 若找到 quote 字元，檢查其前個字元為 X 或 C（在 SIC 中，X 字元後面接 quote 表示十六進位數字、C 字元後面接 quote 表示字串）；往後讀，並存入 TKBUF，直到遇到下個 quote 字元，依照前面條件判斷 TKBUF 該存入編號 6 (Integer/Real)或編號 7 (String)的 Table，並記錄 Token 資訊並存至 LBUF。

2-3. 若找到其他分隔符號，則檢查 TKBUF 大小是否大於 0，是則將 TKBUF 內容存入至 Table，紀錄 Token 資訊；再將找到的分隔符號比對 Table，紀錄 Token 資訊存至 LBUF。

3. 若前個步驟找到的字元為分號（在 SIC 中，分號字元後為註解），則往後讀取字元，直到找到換行或 EOF。
4. 重複步驟 1，直到前述過程中讀取到 EOF，則程式結束。

取得 Token 資訊流程：

1. 將 TKBUF 內容依序比對 Table 編號 1~4，若匹配到相同字串，則回傳該 Token 所在 Table 編號以及該 Token 於該 Table 內的 index，此兩個數字即為一個 Token 資訊。
2. 若 Table 編號 1~4 內無匹配，則表示該 Token 為一個 Symbol，於編號 5 的 Table (Symbol)內嘗試找尋是否存在該 Token，否則將 TKBUF 利用 Hash Function 計算 Hash Value 並存入該 Hash Table，回傳編號 5 表示該 Token 種類以及該 Token 於 Table 內的 index。

三、資料結構

1. Instruction、Pseudo、Register、Delimiter Table
2. Symbol、Integer/Real、String Table (Hash Table)
3. Token Buffer (TKBUF，讀取原始碼字元時，作為 Token 暫存區)
4. Line Buffer (LBUF，存放一行原始碼切割成 Tokens 的暫存區)

肆、未完成的功能

Lexical Analysis 大致已完成。