**系統程式Final Project - SIC Assembler**

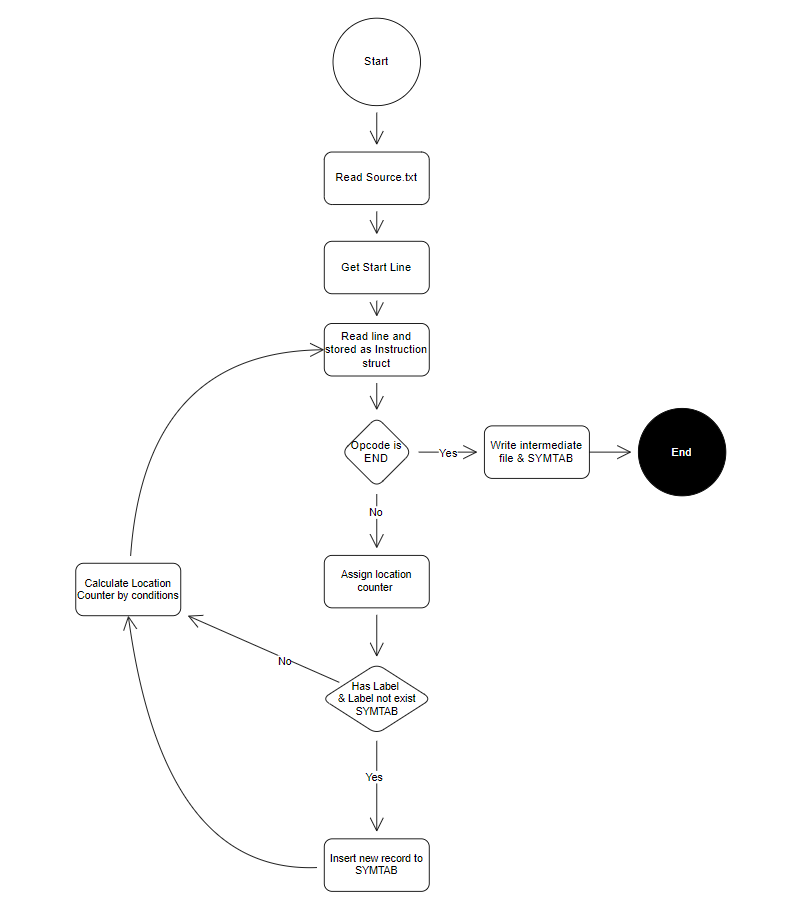
5110056005 劉祈廷

**1. 執行環境**

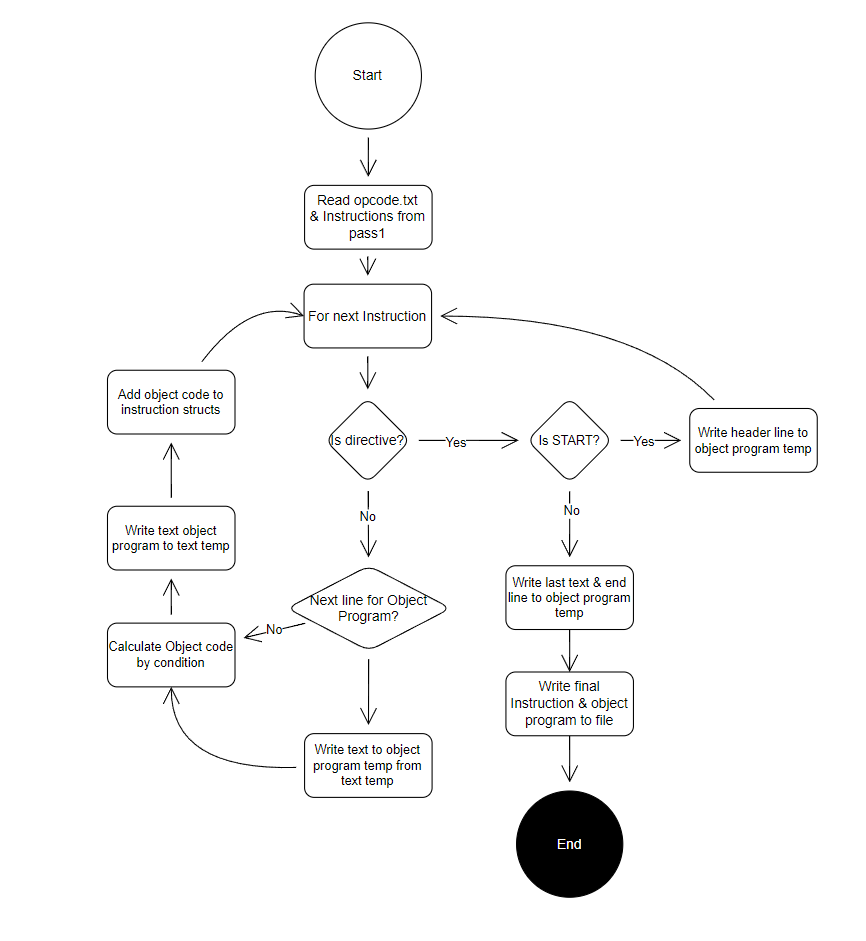
* Compiler: TDM-GCC 9.2.0
* Standard: C++11以上
* 執行時請將 source.txt, opcode.txt 與 .cpp 放在同一階層
* Output:
  + pass1-instructions\_list.txt
  + pass1-symbol\_table.txt
  + pass2-instructions\_list.txt
  + pass2-object\_program.txt

**2. 程式流程**

**Pass1**



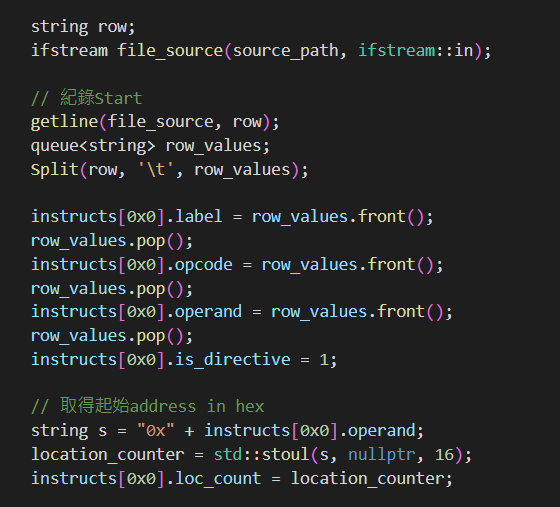
**Pass 2**



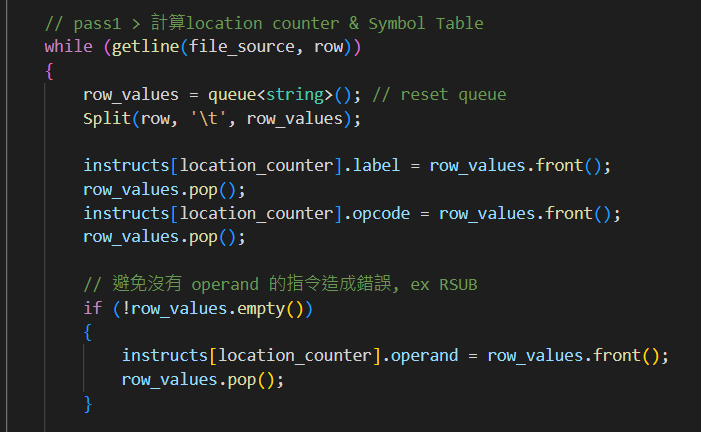
**3. 程式重點**

**Pass1**

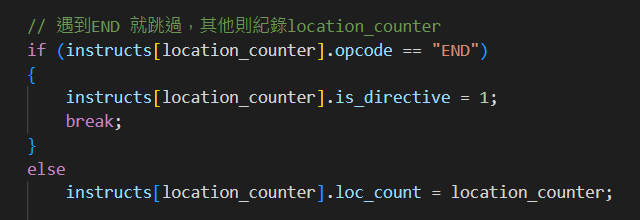
1. 讀取source並將header line 保存至struct 中，初始化 location counter



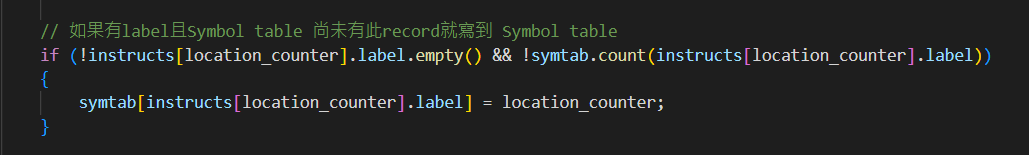
1. 輪循每一行指令，先將所有instruction 轉為 struct 方便後續處理



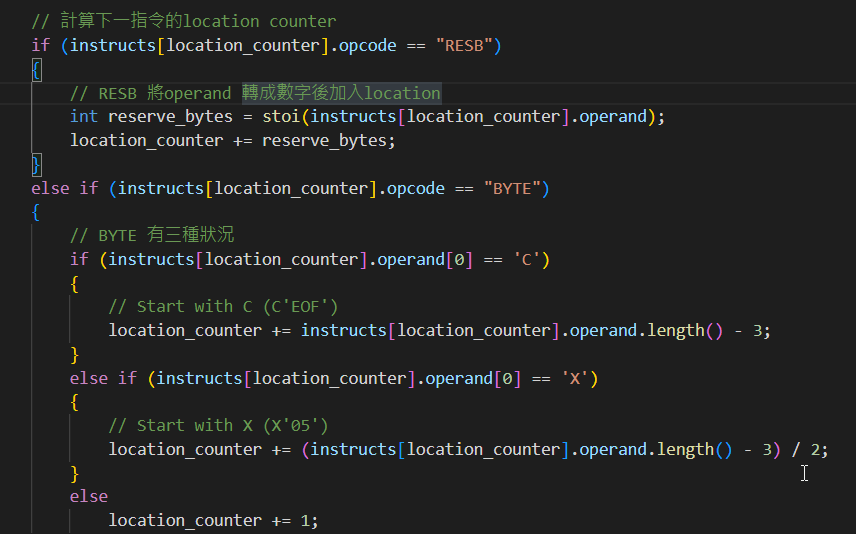
1. 遇到END 就結束pass1，否則將當前的location counter 記錄到 struct 中



1. 判斷有Label 且 Label 不存在於SYMTAB時寫入symbol table

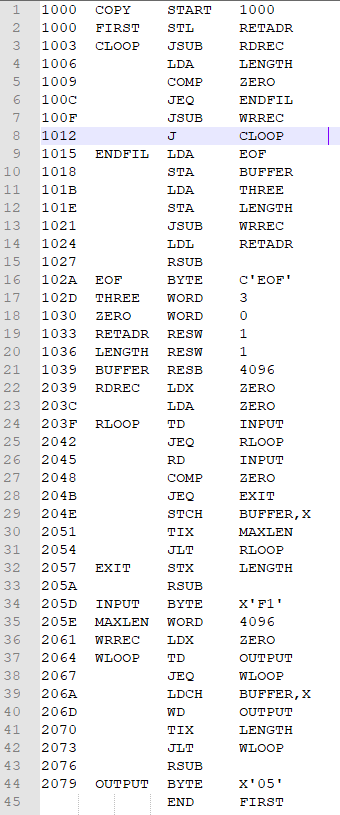


1. 計算下一個指令的location counter

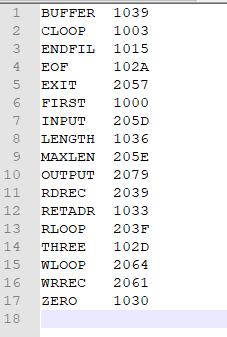


1. 最後寫入檔案，結束pass1，pass1 輸出結果如下

* pass1-instructions\_list.txt

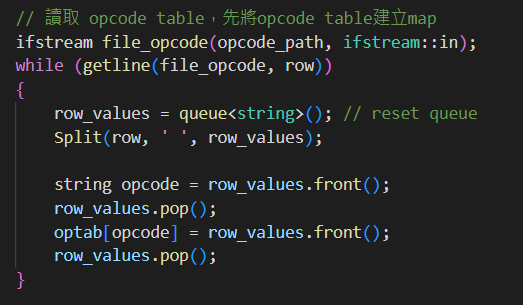


* pass1-symbol\_table.txt (我的SYMTAB是用Label 字母順序排序)

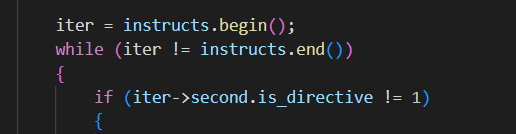


**Pass 2**

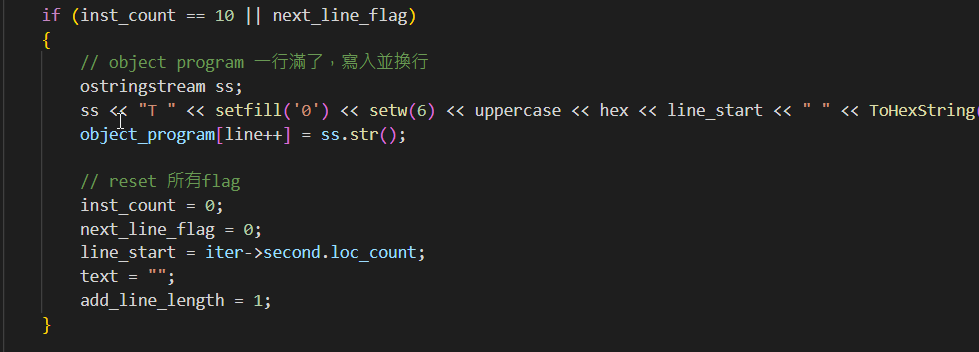
1. 讀取opcode.txt



1. 輪循每一個指令，先判斷是不是 directive



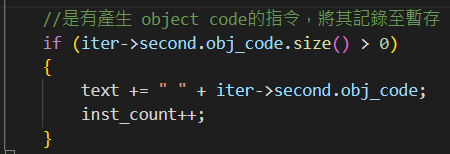
1. 如不是 directive，先判斷object program是不是需要換行了，是的話將暫存的text 加入object program 的暫存中，待後續寫入檔案



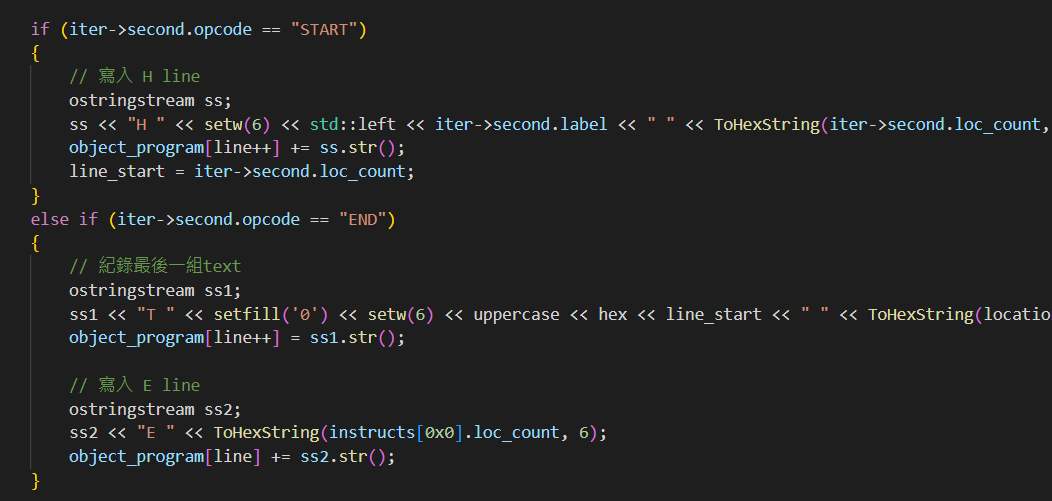
1. 後面依據各operator的特性轉換成 object code，將object code 存回struct 中



1. 如有產生object code，將object code 寫入text 暫存，待換行時整行一同寫入object program的暫存

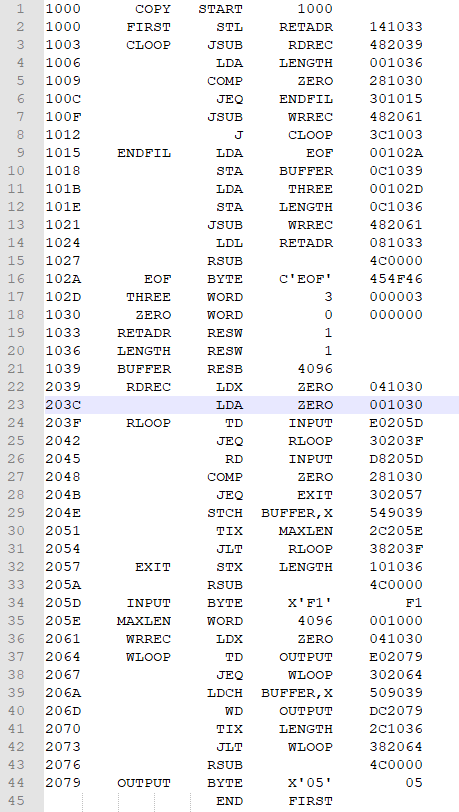


1. 如果指令是directive，START 的情況直接header line 寫入object program的暫存，END的情況要連著最後一行Text line 把End line寫入 object program 暫存

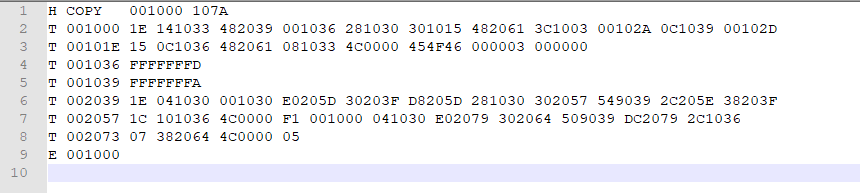


1. 最後將instruction struct 和object program temp 寫入檔案，結果如下

* pass2-instructions\_list.txt



* pass2-object\_program.txt



1. 結束