**開發平台**

* Dev C++
* 4G
* Intel i5-6198DU CPU

**使用開發環境**

* Windows 7
* Dev C++
* 使用[MinGW](https://zh.wikipedia.org/wiki/MinGW" \o "MinGW)及[GDB](https://zh.wikipedia.org/wiki/GDB" \o "GDB)作為編譯系統與除錯系統
* 整合式開發環境（IDE）

**使用的程式語言**

C++  
**所選擇的組合語言(x86 , SIC )**

SIC和SIC/XE  
**說明你的程式設計(功能，流程，使用的data structure)**

**使用的data structure:**

* struct table:用來存7個table，table型別皆為vector<string>，分別是Table1(Instruction)、Table2(Pesudo&Etra)、Table3(Register)、Table4(Delimiter)、Table5(Symbol)、Table6(Integer/Real)、Table7(String)
* struct tk: 用來存切好的token資料，3個欄位為string name、int val和int tb，分別存切完的token名、token在所屬table的位置和所在的token編號
* class tablefc:將會用到的function做封裝，內有build1to4()、read(string & fname)、cut()、sum(string str)、hash( string str, int tb)和issame(int & val, string key)，共6個function
* vector<string> instr:用來放輸入的原始程式指令
* vector<tk> altk:用來存所有切完的token資料

**功能:將輸入的原始程式切token**

* void build1to4():將輸入檔1、2、3、4table內資料放入和其檔名數字編號對應的4個vector<string>型別的 Table1、Table2、Table3和Table4，並設Table5、Table6和Table7大小為100(範圍為0~99)
* void read(string & fname):分行讀入user輸入的原始程式檔，並一一放入vector<string> instr
* void cut():將instr內原始程式分行做切token
* int sum(string str):計算並回傳傳入變數str內各字元的ASCII總和(十進位)
* void hash( string str, int tb):hash function，將sum(string str)回傳的總和mod 100後所得(為str在table內的位置)放入編號為tb的table相應位置內，若遇碰撞則存入下一個位置，再遇碰撞以此類推
* int issame(int & val, string key):檢查傳入字串key是否重複出現，是，val設為在table內所在位置，並回傳table編號，否，val設為0，並回傳0
* main():主程式

**流程**

順序

1. build1to4()

先將資料存入1、2、3table，並設定5、6、7table size

1. read(string & fname)

讀入輸入檔內原始程式

1. cut()

切token，遇到Space/Tab/Enter(white space)/Delimiter，到各table檢查是否為已設定的指令、符號等…，是則建立token，遇到Delimiter，到各table檢查是否為已設定的指令、符號等…，是則建立token(可能有一個或兩個)，查table沒此token，token為symbol、interger/Real、String之一，以mod 100 hash function放入table，遇碰撞則index+1

1. ofstream

將結果放入輸出檔，格視為一行source，一行為output出來的結果