1. **開發平台 :** Dev-C++ 5.11
2. **開發環境 :** Windows 10
3. **使用程式語言 :** C++
4. **所選組合語言 :** SIC
5. **程式各功能說明 :**

int FindNStore( int ch, bool isSame )

用來給readRecords( )呼叫，每次呼叫時傳入一個讀入字元的值，並且在這個function中放入我的token list中。

bool readRecords(vector<TokenType> &cSet)

由檔案中，一個字元一個字元讀取出，做出該字元的類型判斷後，呼叫FindNStore( ) (上式)，將該字元放入token list中，執行完此function，能將所有token完整切出，token list中的資料完整完成。

void introMsg(void)

純粹作為輸出標題訊息使用。

bool readTable( int tableNum, int cur, vector<TokenType> &cSet )

讀取1、2、3、4 table檔，依照傳入之參數決定讀取的table，此function功能在於找出token所在的座標(table, location)，找到所求後，進入token list設定該token之座標。 若沒有找到，則return false 使該token進入待創table的名單中(即5、6、7 table)。

void createATable1( int , bool, vector<TokenType> &cSet )

將所有待歸類的token進入此function，只處理5、6、7 table的候選者，在仔細判斷後，加上hash function並放入5、6、7 table，同時，進入token list設定該token之座標。

void createATable2( )

未完成之功能。

void putAllRecords( vector<TokenType> &cSet)

輸出所有結果至螢幕(包含token及其座標)，作為檢查與debug時使用。

void writeRecords( )

將原始程式碼及token座標輸出寫檔成.txt。為最終結果。

1. **程式使用之Data Structure :**

Token List使用Vector，每個節點擁有三項資料: token本身、所在table以及在table內的位置邊編號。

另有一List，使用Vector，每個節點只有一項資料: 待放到5、6、7 table的token在token中的位置，此List包含所有未在1、2、3、4 table者，但只告知其在真正 Token List中的位置，存取仍是由Token List。

Table 5、6、7使用大小為100的Array，每個節點中有兩項資料: token名稱及節點編號。

1. **程式使用流程 :**

執行程式後，使用者於螢幕看見Greeting Message，隨後可輸入要讀取並解析的資料檔檔名，切好token後程式將自動找出各個token的在table中的位置，並寫出檔案包含原程式碼與當中toke的座標。

1. **未完成之功能 :**

void createATable2( )

原先要將5、6、7 table也寫成table輸出，後來因故作罷。

並且，在讀檔階段，程式也無法因為錯誤而循環要求使用者重新輸入檔名，而是告知無此檔案，隨後退出(然此點修改相當容易)。