**SIC Assember**

**資訊二丙 82 D0588813 曾玉鳳**

**資訊二丙 54 D0542900 董育汝**

**資訊二丙 18 D0511203 陳品樺**

**資訊二丙 21 D0511305 范瑋軒**

**Assembler 開發語言及平台 :** 使用Python

**處理步驟 :**

Step1 : 將檔案讀入，使用Python分割字串的特定語法，將字串分割並存取，

如果中間讀到tab，則將tab error寫入Lisfile檔並停止程式

Step2 : 區分出虛擬指令，我們使用if-else來分別出START、RESB、RESW、BYTE、WORD及END，計算Location時以10進位運算，算完再轉至16進位，如遇到label，則另外存至sym={ } (dictionary)，。

遇到WORD轉為16進位，直接計算WORD及BYTE的object code。

若遇到有X的opcode，則切割至' , '前之opcode，並放入該位址第一位加8，轉成16進位，放入該位址後3位。

Step3 : 將其餘的object code編出，

Step4 : 將內容讀出寫成LISFILE.txt檔並輸出，若有錯誤，亦會顯示於內

Step5 : 將內容讀出寫成OBJFILE.txt檔並輸出。

輸出H卡片

將上面編好的object code依序寫入T卡片，若長度不滿30，則繼續

寫入若超過或下行object code為空，則換一張T卡片繼續編寫

如果遇到「.」、空字串(第一行)，表示是註解、RESB、RESW，省略不寫入

如果遇到END，則先將前一行T卡片未完成字串輸出，再輸出E卡片

**輸入格式:**

不可使用tab

英文大、小寫皆可輸入

Label、指令及operand起始位置有固定要求(1-9，10-17，18- )

不一定要有label

註解可新的一整行，不可放於後面

**可處理的addressing modes和assembler directives:**

addressing modes : Direct addressing

assembler directives: START、END、WORD、BYTE、RESW、RESB

**Function :**

hex( ) : 十進制轉16進制

ord( ) : 將字元轉為ASCII

append ( ): 將東西新增在列表後方

insert( ) : 將指定對象插入列表的指定位置

zfill( ) : 將字串前方補0，使之達指定字串長度

upper( ) : 將小寫轉大寫

split( ) : 通過指定分隔符對字串進行切割

strip( ) : 去除字頭字尾指令字元(默認為空白鍵)

math.ceil( ) : 取整數，無條件進位

**data structures :**

**sic list :** 很多個list，每個list包含多個小list

[ [[10進位 , 16進位] , label , opcode , operand , object code] ,[…..] , ……. ]

Ex. [[[8192, '2000'], 'HW1SIC', 'START', '2000', ''],

[[8192, '2000'], 'FIRST', 'LDA', 'AA', '00203C'],

[[8195, '2003'], '', 'ADD,X', 'BB', '18A03F'],

[[8198, '2006'], '', 'STA,X', 'CC', '0CA042'],……]

**sym dict. :** {`label名稱’: `location’,…… }

Ex. {'AA': '203C',

'BB': '203F',

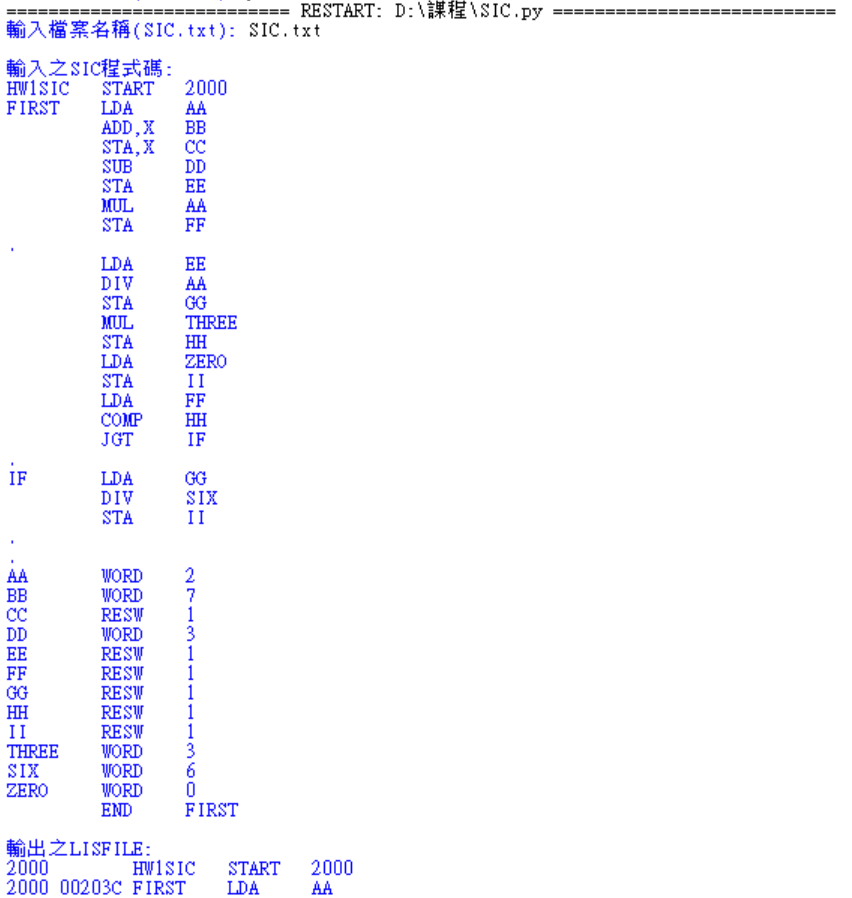
'CC': '2042',

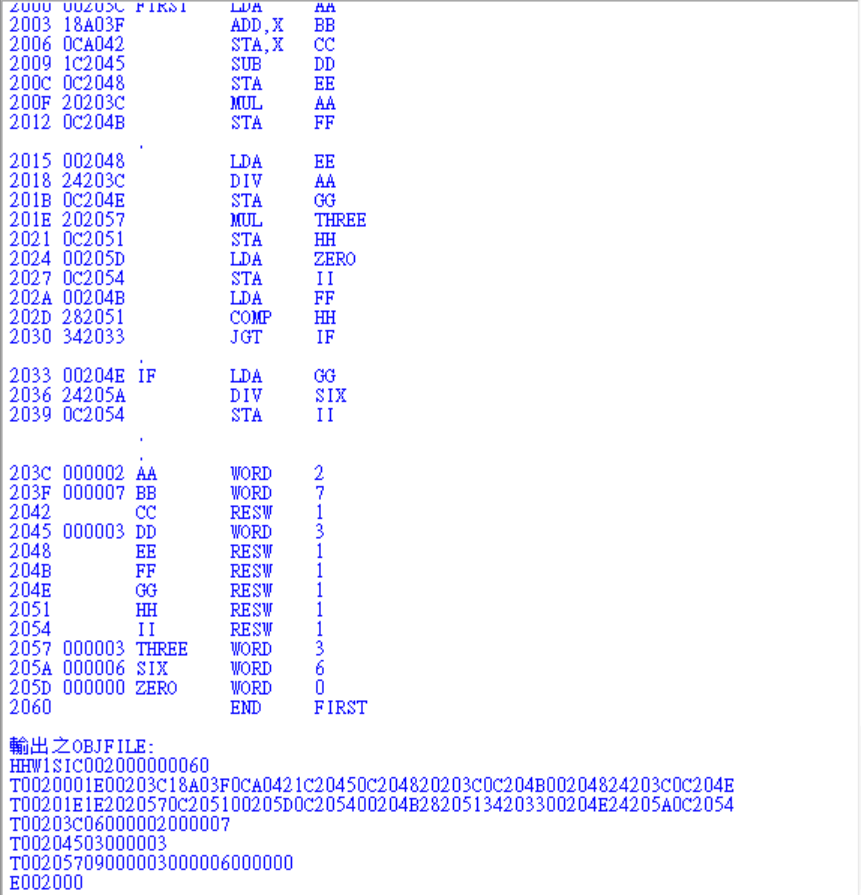
'DD': '2045',

'EE': '2048',

….. }

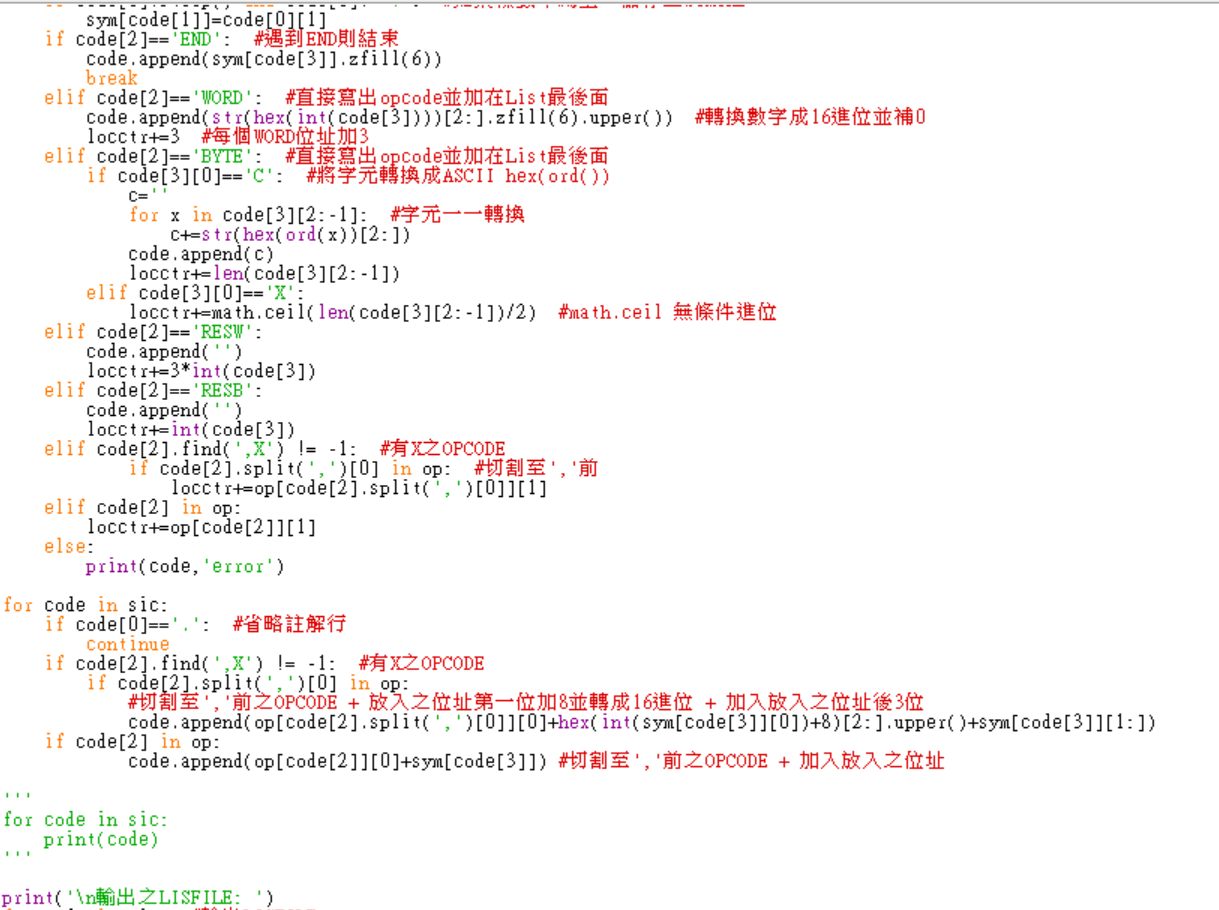
**輸出格式範例:**





**程式碼:**

****

****

****