**SIC Assembler**

**班級:資訊二甲 學號:D0441262 姓名:鄭藍慈**

**班級:資訊二甲 學號:D0440669 姓名:林羿汶**

**班級:資訊三甲 學號:D0308939 姓名:郭怡君**

***￭程式步驟:***

(使用DEV-C++)

**Step 1:**

先將檔案讀入並切割每一行字串，將Label、Opcode、Operand分類放入Struct中。

**Step 2:**

區分出虛擬指令，我們使用if-else來分別出START、RESB、BYTE、及END，依照Opcode計算Location。

**Step 3:**

找出Label的Location並存入struct，之後要計算Object code 所要使用到的。

**Step 4:**

計算Object code，我們使用if-else來分別各個Opcode。

***￭ data structures***

struct Var

{

char La[10]; //Label

char VLo[5]; //loc (前面的)

};

struct OgAry

{

char Location[5]; //計算Object code的

struct Var Op1;

char Op2[10]; //Opcode

char Op3[10]; //Operand

char Other[50];

char OBcode[10]; //Object code

char EndLo[5];

};

***￭ function***

void CalLocation(OgAry\* ); //計算Location

void CalOBcode(OgAry\* ); //計算 OBcode

void SerLocation(OgAry\* ); //label 的 location

void Strspilt(const char source[], char dest[], const int start, const int end );

＊const char source[] 資料來源 char dest[] 將被修改的陣列

const int start 起始位置 const int end 結束位置

→將分割字串儲存進陣列中

int ChangtoX(char ch[],int plus);

int plus 虛擬指令的BYTE

→把字串(16進制)轉換成10進制 + 虛擬指令的BYTE

Ex : 1000->4096

int ChangtoD(char ch);

→將16進制轉換成10進制

Ex : A->10 或 F->15

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

START、END、WORD。

Direct addressing。

***￭可處理的格式 :***

**-**不可用Tab

-英文大小寫都可

-只要 label、Opcode、Operand之間有空白都可以。

-不一定要有label