本次Project必須使用的資料結構( Data Structure )

※ 一定要使用此資料結構，否則0分計算( 期中考也會考 )

enum TokenType { IDENTIFIER = 34512, CONSTANT = 87232, SPECIAL = 29742 } ;

typedef char \* CharPtr ;

struct Column {

int column ; // 此token(的字首)在本line有出現於此column

Column \* next; // 下一個此token(在本line)有出現的column

} ; // Column

typedef Column \* ColumnPtr ;

struct Line { // struct Line 記錄了4種資訊

int line ; // 此token放在哪一 line

ColumnPtr firstAppearAt ; // 指到此token第一次出現在此line的哪一column

ColumnPtr lastAppearAt ; // 指到此token最後出現在此line的哪一column

Line \* next; // 同一 token 下一次出現在哪一 line

} ;

typedef Line \* LinePtr ;

struct Token { // Struct token 紀錄了4種資訊

CharPtr tokenStr ; // tokenStr 放你切下來的token

TokenType type; // type 紀錄此token是屬於哪一個case

LinePtr firstAppearOn ; // 此 token 第一次出現在哪一 line

LinePtr lastAppearOn ; // 此 token 最後出現在哪一 line

Token \* next ; // 指向下一個token

} ; // struct Token

typedef Token \* TokenPtr ;

TokenPtr gFront = NULL, gRear = NULL ; // 分別指向Token串列的頭跟尾

typedef char Str100[ 100 ] ; // 此型別是打算用來宣告一個陣列變數、以供讀字串之

用

int gLine = 1 ; // 「下一個要讀進來的字元」所在的line number

int gColumn = 1 ; // 「下一個要讀進來的字元」所在的column number

// 以上是夏老闆在程式之中的global area所放的程式碼。

至於Token這個型別之中的tokenStr要如何用？

夏老闆的想法是這樣，首先，GetToken()的寫法應該是如下

(但沒說一定要這樣，這裡只是要示範tokenStr的"製作"方式)：

CharPtr GetToken( ... ) { // GetToken()可能會有一些參數

Str100 tokenStr = { 0 } ; // 初始化

CharPtr returnTokenStr = NULL ;

returnTokenStr = new char[ strlen( tokenStr ) + 1 ] ;

strcpy( returnTokenStr, tokenStr ) ;

return returnTokenStr ;

} // GetToken()

任何function若要呼叫GetToken()(如果GetToken()是如上宣告的話)，

就應該做(類似)如下的動作：

Token newToken ;

newToken.tokenStr = GetToken( ... ) ; // now, tokenStr指向一個token的相對應

string

// 以下只是示範給你看你在答題時、你的function可以如何寫。

void PrintTokenInfo( int i ) {

// 印出 gFront 指向的 Token 串中第i個token的相關資訊

TokenPtr current = gFront ;

for ( int j = 0 ;

current != NULL && j < i ;

j ++ )

current = current->next;

printf( "%s ： ", current.tokenStr ) ;

LinePtr lineWalk = NULL ;

ColumnPtr columnWalk = NULL ;

for ( lineWalk = current.firstAppearOn ;

lineWalk != NULL ;

lineWalk = lineWalk->next )

for ( columnWalk = lineWalk->firstAppearAt ;

columnWalk != NULL ;

columnWalk = columnWalk->next )

printf( "( %d, %d ) ", lineWalk->line, columnWalk->column ) ;

printf( "\n" ) ;

} // PrintTokenInfo()