一、 Design abilities

在本次作業中，設計目標是實作 c 語言的 scanner，將取出原始碼中 的 token，並加以區分，以利於之後 parser 的實作，而參考的標準為 c99 文 件。 本次實作的 scanner 可將得到的 token 分為以下九個類別，分別是  
key、 id、 op、 punc、 integer、 double、 char、 sci 和 string，並且也有將  
comment 部份做過濾。除此之外， 還有實作 pragma 功能，一共有四項參 數，分別為 source on、 source off、 token on 和 token off， source 用來控制 是否顯示原始碼， token 則控制是否顯示 token 分類結果。 二、 Design decisions 本部份將詳述 scanner 的實作方式，包含 token 的分類、 comment 的 處理、 pragma 實作以及 error handling 的判斷。 首先， 將根據實作的順序， 介紹 token 的分類和相對應的 error  
handling 方式。 在 Keyword 部份十分單純，只需包含 spec 中所列出的項目和 stdio.h 中的 function，一旦比對到符合上述的字串，即可歸類至 key token。 在 Operator 和 Punctuation character 部份，實作上也較為簡便，如同  
Keyword，只需包含 spec 中所列出的項目，即可完成實作。 在 Char 部份，設計的方法是偵測以「 '」開頭和結尾，並且中間只包含 一個字元的字串，需特別注意的是'\t'和'\n'也是合法的 Char，因此這邊有 針對這兩種情形特別做判斷。 在 String 部份，設計的方法是偵測以「 "」開頭和結尾所組成的字串， 由於 String 需要實作 error handling，我所做的判斷為， String 開頭和結尾中 只能由非「 "」、「 \」 和「 \n」的字元組成，或者由「 \.」 (.代表除了 \n 以外 的所有字元)，如此就能過濾 String 的所有可能錯誤。 在 Scientific notation 部份，偵測由 integer 或 double 開頭且後方連接  
(e|E)(+| -)?[0-9]+的字串，其中 (+| -)?所代表的是 Scientific notation 中次方為 正號或負號或沒有正負號的三種情形。 在 Double 部份，偵測的內容為(數字集合). (數字集合)、 (數字集合). 和.(數字集合)三種情況， regular expression 為([0-9]+"."[0-9]\*)|([0-9]\*"."[0-  
9]+)。 error handling 方面，處理的是有一個小數點以上的情況，例 如:1.2.3.4.5，將列為錯誤內容。 在 Integer 部份，偵測的內容為(數字集合)。 error handling 方面，則是 處理 0 開頭的數字集合，也就是將 0(數字集合)列為錯誤。 在 Identifier 部份，偵測的內容為由數字或大小寫字母或「 \_」或「 $」 所組成的字串。 error handling 方面，則是處理由數字開頭的 Identifier，例 如: 1abc\_de， 即會列為錯誤。 以上就是所有 token 的分類，但除此之外還有以下三種可能的合法  
input 字串，分別為「 \n」、「 \t」和空白字元。若是有任何 input 字串 無法 被上述的任一類別所接收，則將其視為 error，例如:「 ?」 和「 "」 即屬於此 類。 接下來將介紹 comment 的處理， 我利用 start condition 來進行實作， 共設立了兩個 exclusive state(c1、 c2)，分別處理「 /\*……\*/」 和「 //」 兩種註 解方式。在前者的部份，偵測到「 /\*」 即進入 c1 state 中，直至偵測到 「 \*/」 才離開 c1，並繼續偵測前述的 token。在後者的部份，則為偵測到 「 //」 即進入 c2 state 中，直至偵測到「 \n」 才離開 c2。 最後， 在 pragma 部份，同樣是運用 start condition 來進行實作，共設 立了三個 exclusive state(p、 psrc、 ptok)，用來處理 source 和 token 兩類  
pragma。判斷的流程為，若偵測到「 #pragma 」，則進入 p state，進入後 則判斷接續字串為「 source」 或「 token」，並分別進入 psrc state 和 ptok  
state，若接續字串非上述兩者，則為 error。在 psrc state 和 ptok state 中， 則偵測「 on」 和「 off」，並該改程式中的 output flag，同理偵測到非「