

姓名:張智超
學號:10230035

a. How to excute your code?

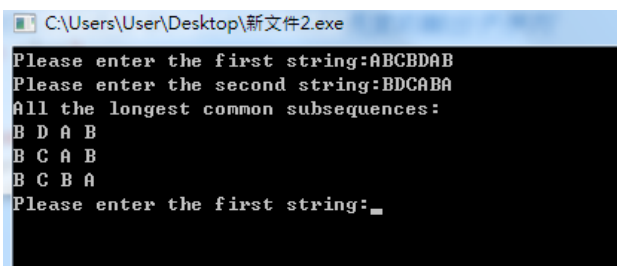
使用 C 語言在 Dev C++的環境下實作。

b. The procedure of your program.

首先使用 scanf 讀入兩個字串，建立二維的 array 來記錄搜尋方向，比較字串是否相符，自底向上進行遞推計算記錄序列 X 和 Y 的 LCS 的長度，並確定二維 array 中的搜索方向，最後參考課本提供之演算法列印出所有共同的字串出來。

```
PRINT-LCS( $b, X, i, j$ )
1  if  $i = 0$  or  $j = 0$ 
2    then return
3  if  $b[i, j] = "\diagdown"$ 
4    then PRINT-LCS( $b, X, i - 1, j - 1$ )
5    print  $x_i$ 
6  elseif  $b[i, j] = "\uparrow"$ 
7    then PRINT-LCS( $b, X, i - 1, j$ )
8  else PRINT-LCS( $b, X, i, j - 1$ )
```

使用課本之例子跑出之結果。



```
C:\Users\User\Desktop\新文件2.exe
Please enter the first string:ABCBADAB
Please enter the second string:BDCABA
All the longest common subsequences:
B D A B
B C A B
B C B A
Please enter the first string: _
```

c. How you use the dynamic programming approach?

參考課本提供之演算法

```
LCS-LENGTH( $X, Y$ )
1   $m \leftarrow \text{length}[X]$ 
2   $n \leftarrow \text{length}[Y]$ 
3  for  $i \leftarrow 1$  to  $m$ 
4    do  $c[i, 0] \leftarrow 0$ 
5  for  $j \leftarrow 0$  to  $n$ 
6    do  $c[0, j] \leftarrow 0$ 
7  for  $i \leftarrow 1$  to  $m$ 
8    do for  $j \leftarrow 1$  to  $n$ 
9        do if  $x_i = y_j$ 
10           then  $c[i, j] \leftarrow c[i - 1, j - 1] + 1$ 
11               $b[i, j] \leftarrow "\diagdown"$ 
12           else if  $c[i - 1, j] \geq c[i, j - 1]$ 
13              then  $c[i, j] \leftarrow c[i - 1, j]$ 
14                   $b[i, j] \leftarrow "\uparrow"$ 
15           else  $c[i, j] \leftarrow c[i, j - 1]$ 
16                   $b[i, j] \leftarrow "\leftarrow"$ 
17  return  $c$  and  $b$ 
```

d. The time complexity analysis of your program.

這個 algorithm 主要有兩層 loop，out loop 次數為 m 次，inner loop 次數為 n 次，algorithm 的時間複雜度為 $T(n)=O(m*n)$ 。