

目錄

第一章	課後習作解答	482
第二章	課後習作解答	483
第三章	課後習作解答	485
第四章	課後習作解答	488
第五章	課後習作解答	491
第六章	課後習作解答	495
第七章	課後習作解答	496
第八章	課後習作解答	499
第九章	課後習作解答	502
第十章	課後習作解答	508

第一章 課後習作解答

一、選擇題

題號	1	2	3	4	5	6	7
答案	B	C	D	A	B	A	C

二、實作題

《Ans 1》參考範例「各章節習題實作\Lab_01_01」

《Ans 2》參考範例「各章節習題實作\Lab_01_02」

《Ans 3》參考範例「各章節習題實作\Lab_01_03」

《Ans 4》參考範例「各章節習題實作\Lab_01_04」

第二章 課後習作解答

《Ans 1》演算法描述解決問題的方法是以程序式的描述為主，讓「人」一看就知道是怎麼一回事，所以表達的對象以人為主，要能閱讀。大部分的演算法都能夠利用程式流程圖表現，不過並非所有程式流程圖都可用演算方法表示，因為程式流程圖可包含無窮迴路，與演算法的定義相違背。

《Ans 2》參考範例「各章節習題實作\Lab_02_02」

Step 1. 定義函式，以名稱、科目和5科成績為參數。

Step 2. 函式中，輸出名稱。

Step 3. 函式中，以變數total來統計總分。

Step 4. 函式中，算出平均分數。

Step 5. 函式中，加入條件判斷，大於或等於60分者，印出總分和平均分數。

Step 6. 函式條件判斷中，小於60分者，印出不及格和平均分數。

Step 7. 主程式，宣告3個陣列sc1、sc2、sc3來儲存分數。

《Ans 3》參考範例「各章節習題實作\Lab_02_03」

執行次數為5次，由於每次在迴圈裡會把k值會從100000、1000、100、10、1；由於「 $k = 1$ 」沒有大於條件判斷就結束while迴圈。

《Ans 4》

(A) $f(n) = (n^2+1)\log n = O(n^2\log n)$

(B) $f(n) = 8\log\log n = O(\log\log n)$

(C) $f(n) = \log n^2 = 2\log n = O(\log n)$

(D) $f(n) = 4\log\log n = O(\log\log n)$

$$(E) f(n) = n/100 + 1000/n^2 \leq n/100 \text{ (當 } n \geq 1000 \text{ 時)} = O(n)$$

$$(F) f(n) = n! = 1 * 2 * 3 * 4 * 5 \cdots * n < n * n * n \cdots * n \leq n^n \text{ (} n \geq 1 \text{ 時)} = O(n^n)$$

《Ans 5》參考範例「各章節習題實作\Lab_02_05」

《Ans 6》參考範例「各章節習題實作\Lab_02_06」

第三章 課後習作解答

(一)填充題

1. 靜態資料結構又稱為密集串列，它使用連續記憶體空間，儲存有序串列的資料。
2. 動態記憶體配置是指變數儲存區配置的過程是在執行時，透過作業系統提供可用的記憶體空間。
3. 下列敘述宣告了一維陣列，共佔用了20 Bytes記憶體空間，能以sizeof()函式來取得。

```
int score[5] = {81, 63, 92, 65, 78};
```

4. 下列敘述宣告了Ary是二維陣列，能存放12個元素。

```
int Ary[3][4];
```

5. 宣告一個「 $2 \times 2 \times 4$ 」的三維陣列，C語言如何宣告？int ary[2][2][4]
6. 填寫下列敘述的結構名稱：①為字元陣列，②為字元字串。
7. 計算字串長度的函式strlen()，複製全部字串的函式為strcpy()，複製部份字串則是strncpy()函式；使用這些字串函式要匯入string.h標頭檔。

(二)實作與問答

《Ans 1》 參考範例「各章節習題實作\Lab_03_01」

《Ans 2》 參考範例「各章節習題實作\Lab_03_02」

《Ans 3》 參考範例「各章節習題實作\Lab_03_03」

《Ans 4》 (1) 假設A陣列m列x n行，

```
m = 1 - (-100) + 1 = 102, n = 100 - 1 + 1 = 100  
Loc(A[1, 12]) = 100 + (1 - (-100)) * 100 * 4 + (12 - 1) * 4 = 40544
```

(2) 假設A陣列有m列×n行

$$m = 10 - 5 + 1 = 6, \quad n = 20 - (-10) + 1 = 31$$

$$\text{Loc}(A[5, -5]) = 100 + (5 - 5) * 31 * 4 + (-5 - (-10)) * 4 = 120$$

《Ans 5》 因為陣列元素 $A_{3,2}$ 、 $A_{2,3}$ 的第1個索引值 $3 > 2$ ，第2個索引值 $2 < 3$ ，而儲存位址 $1110 < 1115$ ；故可推知此陣列是以行為主（Column-Major）的方式儲存。

$A_{3,2}$ 的位址計算如下：

$$\alpha + (2-1) \times m \times 1 + (3-1) \times 1 = 1110$$

$$\alpha + m = 1108 \text{ ----- (1)}$$

$A_{2,3}$ 的位址計算如下：

$$\alpha + (3-1) \times m \times 1 + (2-1) \times 1 = 1115$$

$$\alpha + 2m = 1114 \text{ ----- (2)}$$

(1) - (2) $\Rightarrow M = 6$ 代入(1)式得 $\alpha = 1102$

《Ans 6》 由 $\text{Loc}(A(3,3)) = 121$ ， $\text{Loc}(A(6,4)) = 159$ ，得知陣列A的配置是以「以行為主」的方式，所以起始位址 α ，單位空間1，則陣列 $A(1:m, 1:n)$

$\Rightarrow \alpha + (3-1)*1 + m*(3-1)*1$ $= \alpha + 2*(1+m) = 121$ $\Rightarrow \alpha + 2 + 2m = 121 \cdots \cdots \textcircled{1}$	$\alpha + (6-1)*1 + (4-1)*m$ $= \alpha + 3m + 5 = 159$ $\Rightarrow \alpha + 3m + 5 = 159 \cdots \cdots \textcircled{2}$
---	--

由①，②式可得 $\alpha = 49$ ， $m = 35 \Rightarrow \text{Loc}(A_{4,5}) = 49 + 4*35 + 3 = 192$

《Ans 7》 二維陣列「以列為主」，利用公式先算出實際陣列的列數及行數：

$$M = 5 - (-3) + 1 = 9 \Rightarrow 9 \text{列}, N = 2 - (-4) + 1 = 7 \Rightarrow 7 \text{行}$$

$$A(1, 1) \text{位址} = 100 + ((1 - (-3)) * 7 * 1) + (1 - (-4)) * 1 = 133$$

《Ans 8》直接代入以列爲主公式：

$$M = 3 - 1 + 1 = 3, N = 4 - 1 + 1 = 4, O = 5 - 1 + 1 = 5$$

$$\text{Loc}(A_{1,2,3}) = 100 + (1 - 1) * 4 * 5 * 1 + (2 - 1) * 5 * 1 + (3 - 1) * 1 = 107$$

《Ans 9》代入以欄爲主的公式：

$$M = 2 - (-3) + 1 = 6, N = 3 - (-2) + 1 = 6, O = 4 - 0 + 1 = 5$$

$$\text{Loc}(A_{2,2,3}) = 300 + (3 - 0) * 6 * 6 * 2 + (2 - (-2)) * 6 * 2 + (2 - (-3)) * 2 = 274$$

第四章 課後習作解答

(一)選擇題

題號	1	2	3	4	5	6
答案	B	C	D	B	A	C

(二)實作與問答

《Ans 1》單向鏈結串列（Single Link List）、環狀鏈結串列（Circular Link List）、雙向鏈結串列（Double Link List）。

《Ans 2》鏈結串列的第一個節點再附設一個「首節點」（Head Node），但是它不儲存任何資訊；有了首節點，表示從它開始就能找到第一個節點，也能藉由它儲存的「鏈結」（或指標）往下一個節點走訪。

由於是單向鏈結串列，欲加到末端的新節點或者要刪除最後一個節點，必須從第一個節點開始，直到最後一個節點才完成加入動作；若有尾節點的指標就能提其效能。

《Ans 3》使用單向鏈結串列可以插入新的項目，有三種方式可供選擇：
①從首節點插入；②從尾節點插入；③從中間的節點插入。不過，我們一定要知道，無論是哪一種方式都是把鏈結的指標指向新的物件。

《Ans 4》優點

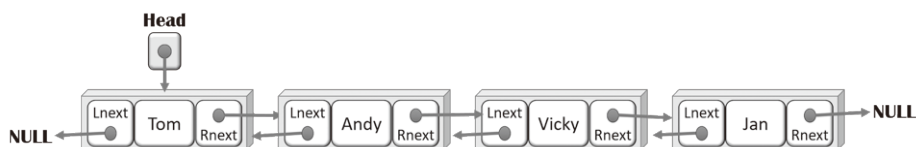
因為雙向鏈結串列有兩個指標分別指向節點本身的前後兩個節點，所以能夠很輕鬆的找到它前後節點，同時從串列中的任一節點也可以找到其他節點而不需經過反轉或比對節點等處理，執行速度較快。另外如果有任一節點的鏈結斷裂，可輕易的經由反方向的串列走訪，快速的完整重建鏈結。

缺點：

由於它有兩個鏈結，所以在加入節點或刪除節點時都得花更多的時間移動指標，且雙向串列較為浪費空間。另外在雙向鏈結串列與單向鏈結串列的演算法中，我們知道雙向串列在加入一個節點時需改變四個指標，而刪除一個節點也要改變兩個指標。不過單向串列中加入節點，只要改變兩個指標，而刪除節點只要改變一個指標即可。

《Ans 5》 參考範例「各章節習題實作\Lab_04_05」

《Ans 6》



《Ans 7》 關於環狀串列的特點，我們大致做出以下的結論：

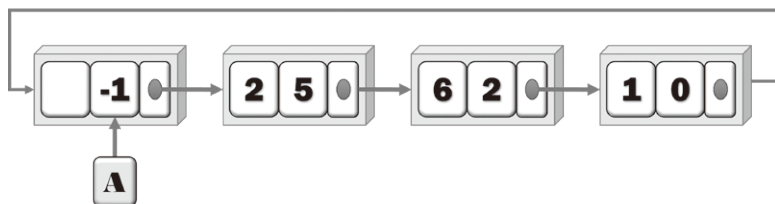
優點：①回收整個串列所需時間是固定的，與長度無關。②可以從任何一個節點追蹤所有節點。

缺點：①需要多一個鏈結空間。②插入一個節點需要改變兩個鏈結。③環狀串列讀取資料比較慢，因為必須多讀取一個鏈結指標。

《Ans 8》 可加一串列首（Head Node）於環狀串列之上，此串列首為



$A = 2 \times 5 + 6 \times 2 + 1$ ，可表示如下：



第五章 課後習作解答

一、填充題

1.	先進後出、pop、push
2.	$-A/*B+CDE$ 、 $ABCD+*E/-$
3.	運算子 + 運算元、運算元 + 運算子
4.	ISP堆疊內優先權、ICP輸入優先權
5.	由右而左、「)」(右括號)、由左而右、「(」(左括號))
6.	4, 5

二、實作與問答

《Ans 1》遞迴呼叫、副程式的呼叫、CPU的中斷處理 (Interrupt Handling)、中序法轉換成後序法、堆疊計算機 (Stack Computer)

《Ans 2》CREATE：建立一個空堆疊

PUSH()：從頂端推入資料，並傳回新堆疊

POP()：刪除頂端資料，並傳回新堆疊

PEEK()：查看堆疊項目，回傳其值

IsEmpty()：判斷堆疊是否為空堆疊，是則傳回true，不是則傳回false

《Ans 3》參考範例「各章節習題實作\Lab_05_03」。

《Ans 4》

$(A+B) * D + E / (F+A*D) + C$	前序++*+ABD/E+F*ADC
-------------------------------	-------------------

$$((((A + B) * D) + (E / (F + (A + D)))) + C)$$

$(A+B) * D + E / (F+A*D) + C$ 後序： $AB+D*EFAD*+/+C+$

$$((((A + B) * D) + (E / (F + (A + D)))) + C)$$

《Ans 5》

$-A* / +BC-DEF$ 前序式： $A-B+C / (D-E) * F$

$$(-A) (* (/ (+B) C) (-D) E) F$$

$AB*CD+E/-$ 後序式： $A*B-C+D/E$

$$(A(B*)) (C(D+)) (E/-)$$

《Ans 6》 $A/B+(C+D)*E-A*C$ 轉為前序式

讀入字元	堆疊內容	輸出
C	Empty	C
*	*	C

讀入字元	堆疊內容	輸出
A	*	AC
-	-	*AC
E	-	E*AC
*	*_	E*AC
)	* -)	E*AC
D	* -)	DE*AC
+	* -) +	DE*AC
(	* -	+CDE*AC
+	+ -	*+CDE*AC
B	+ -	B*+CDE*AC
/	/+ -	/B*+CDE*AC
A	/+ -	AB*+CDE*AC
None	Empty	-+/AB*+CDE*AC

A/B+(C+D)*E-A*C轉為後序式

讀入字元	堆疊內容	輸出
A	Empty	A
/	/	A
B	/	AB
+	+	AB/
(	+(	AB/
C	+(	AB/C
+	+(+	AB/C
D	+(+	AB/CD
)	+	AB/CD+
*	+*	AB/CD+

讀入字元	堆疊內容	輸出
E	+*	AB/CD+E
-	+ -	AB/CD+E*
A	+ -	AB/CD+E*A
*	*+ -	AB/CD+E*A
C	*+ -	AB*+CDE*AC*+ -
None	Empty	AB*+CDE*AC*+ -

《Ans 7》所謂「尾歸遞迴」（Tail Recursion）就是程式的最後一個指令為遞迴呼叫，因為每次呼叫後，再回到前一次呼叫的第一行指令就是return，所以不需要再進行任何計算工作，因此也不必保存原來的環境資訊（如參數儲存、控制權轉移）。例如N!的遞迴式。

《Ans 8》參考範例「各章節習題實作\Lab_05_08」。

《Ans 9》參考範例「各章節習題實作\Lab_05_09」。

第六章 課後習作解答

《Ans 1》 CPU的排程，播放器待播放的曲目，列表機待列印的文件、I/O設備的配置。

《Ans 2》 front指標會指向第一個元素，而rear指標則指向最後一個元素。新增元素時rear指標會隨著新增元素來變更位置。當佇列前端第一個元素被刪除時，指標front原本會由第一個位置而指向下一個位置。所以，指標front會隨著前端元素的移除向後方移動。

《Ans 3》 實作佇列，新增（enqueue）、刪除（dequeue）、佇列是否已滿（isFull）、佇列是否空的（isEmpty）、取得佇列前端元素（front）。

《Ans 4》 「雙佇列」（Deque）是「Double-ends Queues」的縮寫，通俗的說法是佇列有兩個開口，可以指定佇列一端來進行資料的刪除和加入。由於佇列有前端（Front）及後端（Rear），皆都允許存入或取出。

《Ans 5》 環狀佇列的操作

	front指標	rear指標
①	4	5
②	4	7
③	4	0
④	5	0
⑤	5	2

第七章 課後習作解答

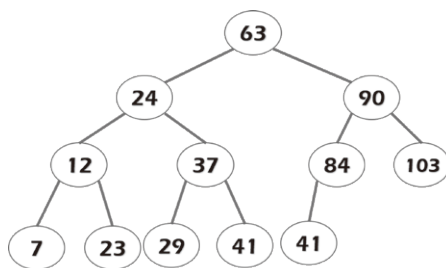
一、填充題

1.	A	A、B、C	D、E、F	兄弟節點	堂兄弟	C、A	B、C、 D、E、F、 G、H
2.	(B)因為環狀串列會造成循環現象，不符合樹的定義						
3.	有序	2					
4.	完滿二元樹		歪斜樹			5.	(A)
6.	空集合	$0 \leq d \leq 2$		沒有次序關係			
7.	葉節點	一個子節點		左、右皆有子節點			

二、實作與問答

《Ans 1》 不是，因為會造成無出口的迴圈。

《Ans 2》 參考範例「各章節習題實\Lab_07_02」。



《Ans 3》 提示：各位可先行假設 n 是節點總數， n_1 是分支度等於1的節點數，可得 $n = n_0 + n_1 + n_2$ ，再行證明。

《Ans 4》 利用數學歸納法證明：

- 當 $i = 1$ 時，只有樹根一個節點，所以 $2^{i-1} = 2^0 = 1$ 成立。

- 假設對於 j ，且 $1 \leq j \leq i$ ，階度為 j 的最多點數 2^{j-1} 個成立，則在 $j = i$ 階度上的節點最多為 2^{i-1} 個。

當 $j = i + 1$ 時，因為二元樹中每一個節點的分支度都不大於2，因此在階度 $j = i + 1$ 時的最多節點數 $\leq 2 * 2^{i-1} = 2^i$ ，由此得證。

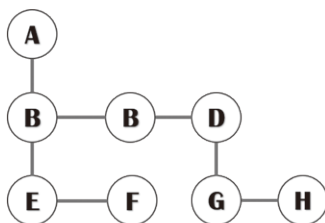
《Ans 5》 中序：A/B**C+D*E-A*C；後序：ABC**/DE*+AC*-；前序：-/A**BC*DE*AC

《Ans 6》 中序：A/B $\uparrow$ C*D-E、前序：-/*A $\uparrow$ BCDE、後序：ABC $\uparrow$ /D*E-

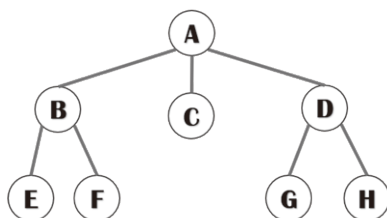
《Ans 7》 樹林的走訪方式則由樹的走訪衍生過來，步驟如下：

中序走訪 (InOrder traversal)	①如果樹林為空，則直接返回。 ②以中序走訪第一棵樹的子樹群。 ③中序走訪樹林中第一棵樹的樹根。 ④依中序法走訪樹林中其它的樹。
前序走訪 (Preorder traversal)	①如果樹林為空，則直接返回。 ②走訪樹林中第一棵樹的樹根。 ③以前序走訪第一棵樹的子樹群。 ④以前序法走訪樹林中其它的樹。
後序走訪 (Postorder traversal)	①如果樹林為空，則直接返回。 ②以後序走訪第一棵樹的子樹。 ③以後序法走訪樹林中其它的樹。 ④走訪樹林中第一棵樹的樹根。
樹林走訪的結果如下：	①中序走訪：EBCDAGHFI ②前序走訪：ABECDFGHI ③後序走訪：EBCDGHIFA

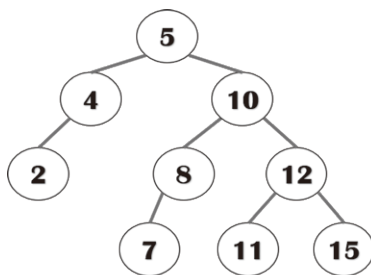
《Ans 8》 這就是樹化為二元樹的反向步驟，方法也很簡單。首先是逆時針旋轉45度，如下圖所示：



另外(ABE)(DG)左子樹代表父子關係；(BCD)(EF)(GH)右子樹代表兄弟關係



《Ans 9》



《Ans 10》如果二元樹的高度為 h ，樹的節點數為 $2h-1$ ， $h \geq 0$ ，則我們稱此樹為「完滿二元樹」(full binary tree)。

如果二元樹的深度為 h ，所含的節點數小於 $2h-1$ ，但其節點的編號方式如同深度為 h 的完滿二元樹一般，從左到右，由上到下的順序一一對應結合。

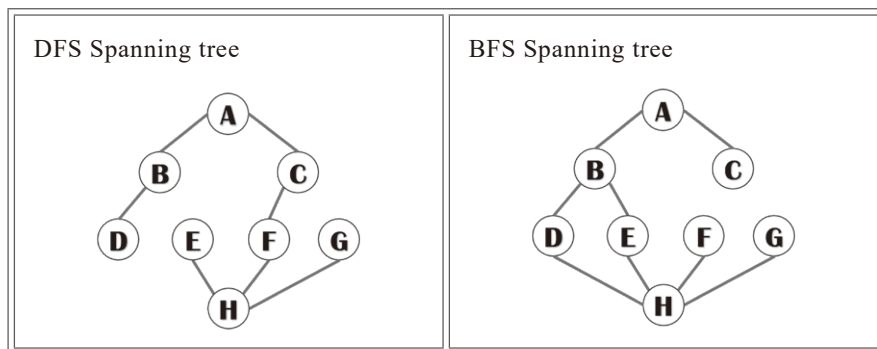
第八章 課後習作解答

一、填充題

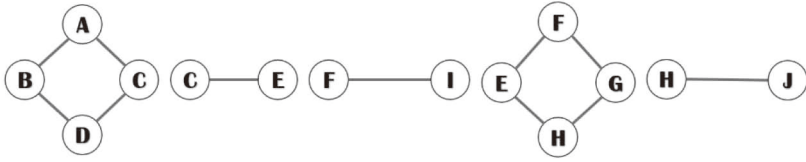
1		2	
一筆畫	偶數分支度	頂點、邊緣、有向、無向	
3		4	
完美圖形		相鄰串列	相鄰矩陣
5		6	
先深後廣走訪	先廣後深走訪	Prim's	Kruskal's

二、問答題

《Ans 1》



《Ans 2》對於一個頂點V，如果將V上所連接的邊都去除所產生的 G' ，如果 G' 最少有兩個連通單元，則稱此頂點V為所謂的「連結點」（Articulation Point）。而一個沒有連結點的圖形，就是「雙連通圖形」（Biconnected Graph）。而這個有4個連結點C、E、F、H，因此並不是「雙連通圖形」。而此圖的連通單元，有下列五種：



《Ans 3》 DFS：先深後廣的走訪順序為：頂點A、頂點B、頂點C、頂點D、頂點E BFS：頂點A、頂點B、頂點E、頂點C、頂點D

《Ans 4》 圖G中的最大加權邊是8，但仍然在MST中。



《Ans 5》

$$A^0 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & \infty & 0 \end{bmatrix} \quad A^1 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix} \quad A^3 = \begin{bmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

《Ans 6》 相鄰矩陣法、相鄰串列法、相鄰多元串列法、索引表格法。

《Ans 7》 一個圖形 $G = (V, E)$ ，存在某一頂點 $v \in V$ ，我們希望從 v 開始，經由此節點相鄰的節點而去拜訪 G 中其它節點，這稱之為「圖形追蹤」。

《Ans 8》 Prim演算法又稱P氏法，對一個加權圖形 $G = (V, E)$ ，設 $V = \{1, 2, \dots, n\}$ ，假設 $U = \{1\}$ ，也就是說， U 及 V 是兩個頂點的集合。然後從 U - V 差集所產生的集合中找出一個頂點 x ，該頂點 x 能與 U 集合中的某點形成最小成本的邊，且不會造成迴圈。然後將頂點 x 加入 U 集合中，反覆執行同樣的步驟，一直到 U 集合等於 V 集合（即 $U = V$ ）為止。

第九章 課後習作解答

一、選擇題

題號	1	2	3	4	5	6	7	8	9	10	11	12	13
答案	D	B	A	D	B	B, C	C	A	D	A	C	B	D

二、實作與問答

1. 請參考範例CH0901.c，如何讓程式完成排序後提早結束？

參考範例「各章節習題實作\Lab_09_01」。

2. 數列「55、234、78、37、165、23、81、46、69、37」繪製出氣泡排序法每一回合由大而小的交換過程，利用程式碼輸出並完成排序。

參考範例「各章節習題實作\Lab_09_02」。

索引	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	說明
原始資料	55	234	78	37	165	23	81	46	69	37'	
A[0] < A[1]	234	55									交換
A[1] < A[2]		78	55								交換
A[2] > A[3]			55	37							不交換
A[3] < A[4]				165	37						交換
A[4] > A[5]					37	23					不交換
A[5] < A[6]						81	23				交換
A[6] < A[7]							46	23			交換
A[7] < A[8]								69	23		交換
A[8] < A[9]									37	23	交換
第1回合得	234	78	55	165	37	81	46	69	37'	23	

繼續第二回合

索引	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	說明
第2回合資料	234	78	55	165	37	81	46	69	37'	23	
A[0] > A[1]	234	78									不交換
A[1] > A[2]		78	55								不交換
A[2] < A[3]			165	55							交換
A[3] > A[4]				55	37						不交換
A[4] < A[5]					81	37					交換
A[5] < A[6]						46	37				交換
A[6] < A[7]							69	37			交換
A[7] < A[8]								37	37'		不交換
第2回合得	234	78	165	55	81	46	69	37	37'	23	

繼續第三回合

索引	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	說明
第3回合資料	234	78	165	55	81	46	69	37	37'	23	
A[0] > A[1]	234	78									不交換
A[1] < A[2]		165	78								交換
A[2] > A[3]			78	55							不交換
A[3] < A[4]				81	55						交換
A[4] > A[5]					55	46					不交換
A[5] < A[6]						69	46				交換
A[6] < A[7]							46	37			不交換
第3回合得	234	165	78	81	55	69	46	37	37'	23	

繼續第四回合

索引	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	說明
第4回合資料	234	165	78	81	55	69	46	37	37'	23	
$A[0] > A[1]$	234	165									不交換
$A[1] > A[2]$		165	78								不交換
$A[2] < A[3]$			81	78							交換
$A[3] > A[4]$				78	55						不交換
$A[4] < A[5]$					69	55					交換
$A[5] < A[6]$						55	46				不交換
第4回合得	234	165	81	78	69	55	46	37	37'	23	完成

3. 數列「185、625、134、47、731、125、42、416」遞增排序以程式碼完成，以手工繪製插入排序法每回合的排序結果。參考範例「各章節習題實作\Lab_09_03」。

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
回合	185	625	134	47	731	125	42	416
1	625	185	134	47	731	125	42	416
2	625	185	134	47	731	125	42	416
3	625	185	134	47	731	125	42	416
4	731	625	185	134	47	125	42	416
5	731	625	185	134	125	47	42	416
6	731	625	416	185	134	125	47	42

4. 將下列資料「185、625、134、47、731、125、42、416」以選擇排序法做遞減排序並繪製出排序過程，此外在第幾回就完成排序？

	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	說明
原始資料	185	625	134	47	731	125	42	416	

527
71
156
264
578
55

(2) 針對每個桶子進行檢測，項目大於「1」還得進行「分配」；將仟位數的「1」號桶子依「百位」進行排序。

桶子	0	1	2	3	4	5	6	7	8	9
仟-1號	1285					1651				

(3) 針對仟位數的「0」號桶子，依「百位」進行排序。

桶子	0	1	2	3	4	5	6	7	8	9
資料	93	156	286			519				
	34		264			527				
	71					578				
	55									

(4) 針對仟位數的「5」號桶子，依「拾位」進行排序。

桶子	0	1	2	3	4	5	6	7	8	9
資料		519	527					578		

(5) 針對仟位數的「2」號桶子，依「拾位」進行排序。

桶子	0	1	2	3	4	5	6	7	8	9
資料			264						286	

(6) 針對百位數的「0」號桶子，依「拾位」進行排序。

桶子	0	1	2	3	4	5	6	7	8	9
資料	34				55			71		93

- (7) 將步驟(6)的資料放回步驟(3)的「0」號桶子；將步驟(5)的資料放回步驟(3)的「2」號桶子；將步驟(4)的資料放回步驟(3)的「5」號桶子。

桶子	0	1	2	3	4	5	6	7	8	9
資料	34	156	264			519				
	55		286			527				
	71					578				
	93									

- (8) 將步驟(2)的資料放回步驟(1)的「1」號桶子；將步驟(7)的資料放回步驟(1)的「0」號桶子。

桶子	0	1	2	3	4	5	6	7	8	9
資料	34	1285			4123					
	55	1651								
	71									
	93									
	156									
	264									
	286									
	519									
	527									
	578									

- (9) 以遞減完成資料的合併。

4123	1651	1285	578	527	519	286	264	156	93	71	55	34
------	------	------	-----	-----	-----	-----	-----	-----	----	----	----	----

第十章 課後習作解答

一、選擇題

題號	1	2	3	4	5	6	7	8	9	10
答案	B	A	D	C	A	C	B	D	B	A

二、實作與問答

《Ans 1》 參考範例「各章節習題實\Lab_10_01」。

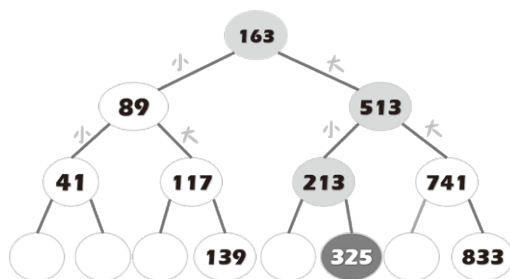
《Ans 2》 參考範例「各章節習題實\Lab_10_02」。

將數列以非遞迴方式撰寫二元搜尋法程式碼來找出Key
「325」，搜尋的過程請以二元樹繪製並簡單說明查找過程的
中間項、最低、最高值的變化。

117, 325, 513, 119, 89, 163, 749, 41, 213, 833

查找過程如下：

```
mid = (0 + 9) // 2 = 4, 為163
325 > 163, 向右
mid = (4 + 9) // 2 = 7, 為513
325 < 513, 向左
mid = 5 + (7 - 1) // 2 = 5, 為213
325 > 213, 向右
mid = (5 + 7) // 2 = 6, 找到key「325」
```



《Ans 3》數列

41, 92, 117, 125, 223, 264, 325, 478, 513, 692, 787

將搜尋數列表示如下：

41	92	117	125	223	264	325	478	513	692	787
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]

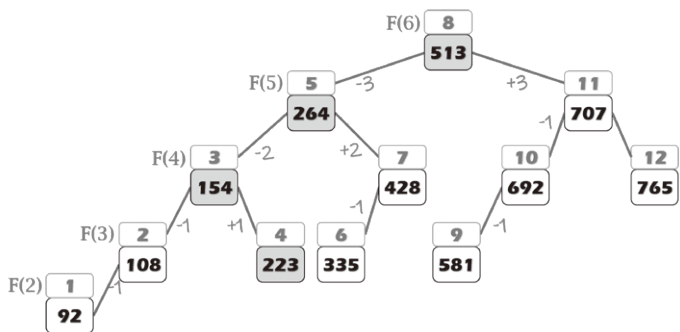
使用「內插法」的搜尋鍵值「513」過程如下：

次數	low	high	mid	key與A[mid]比較	範圍
1	0	10	$0 + \frac{513 - 41}{787 - 41} * 10 = 6$	$513 > 325$	向右
2	$6 + 1 = 7$	10	$7 + \frac{513 - 478}{787 - 478} * 3 = 7$	$513 > 478$	向右
3	$7 + 1 = 8$	10	$8 + \frac{513 - 513}{787 - 513} * 2 = 8$	$\text{mid} = \text{low} = 513$	找到

《Ans 4》數列

92, 108, 154, 223, 264, 335, 428, 513, 581, 692, 707, 765

先產生如下圖的費氏樹



「N = 11」所以費氏樹「Fib(k) - 1 = 12, Fib(k) = 13」，得「k = 7」以費氏樹搜尋key「223」的過程如下：

次數	樹根(r)	子樹(s)	差值(d)	比較	範圍
開始	F(7 - 1) = 8	F(7-2) = F(5) = 5	F(7 - 3) = F(3) = 3	223 < 152	向左
2	r - d = 8 - 3 = 5	s = d = 3	s - d = 5 - 3 = 2	223 < 64	向左
3	r - d = 5 - 2 = 3	s = d = 2	s - d = 3 - 2 = 1	223 > 154	向右
4	r + d = 3 + 1 = 4	s - d = 2 - 1 = 1	d - s = 1 - 1 = 0	223 = 223	找到

《Ans 5》 令雜湊函數為「h(key) = key mod B」，其中「B = 11」為一質數，這個函數的計算結果介於0 ~ 10之間(即0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10這11種可能的餘數)，則

h(345) = 4、h(348) = 7、h(80) = 3、h(119) = 9、
h(83) = 6、h(89) = 1、h(297) = 0

索引	0	1	2	3	4	5	6	7	8	9	10
儲存值	297	89		80	345		83	348		119	

《Ans 6》 利用移動折疊法的原理可得到：
=>743 + 280 + 321 = 1344、
=>1344即為索引位址

《Ans 7》 雜湊表格的大小m = 11（即位址空間），鍵值如下，請以平方

測探來改善碰撞情形：

365, 431, 597, 459, 128, 534, 583, 343, 680, 385

平方測探法

$$h(431) = 431 \% 11 = 2 \rightarrow 3$$

$$h(597) = 597 \% 11 = 3 \rightarrow 4$$

$$h(343) = 343 \% 11 = 2 \rightarrow 6 \rightarrow 0 \rightarrow 7 \rightarrow 5$$

$$h(385) = 385 \% 11 = 0 \rightarrow 1$$

雜湊表如下：

索引	0	1	2	3	4	5	6	7	8	9	10	11
鍵值	583	385	365	431	597	343	534	128	459	680		

