



# 全國高級中等學校 105 學年度商業類學生技藝競賽

## 【程式設計】職種【術科】試卷

崗位編號：\_\_\_\_\_ 姓名：\_\_\_\_\_

各個子題均提供 2 組測試輸入檔，檔名分別是「in1.txt」及「in2.txt」。選手製作的程式，應依序讀入「in1.txt」及「in2.txt」檔，程式執行後，並產生 1 個輸出檔「out.txt」。(即，每個程式讀入 2 個輸入檔，產生 1 個輸出檔。)在輸出檔中，選手應先輸出「in1.txt」產生的結果，再輸出「in2.txt」的結果，兩組結果間用 1 行「空白行」隔開。不影響結果的空白鍵，不列入扣分。若程式執行檔執行結果未依序、不全或無法執行，該子題以零分計算。

題目	子題	題目內容	配分
Problem1	子題 1	計算字數	9
	子題 2	摩斯電碼	11
Problem2	子題 1	網段 ID	13
	子題 2	最大公約數計算	12
Problem3	子題 1	是否為堆積樹(Heap tree)或二元搜尋樹(Binary search tree)	14
	子題 2	後序表示法 (post-order)	15
Problem4	子題 1	最長共同子序列(Longest common subsequence)	10
	子題 2	霍夫曼編碼 ( Huffman Coding )	16

線上評分網址 <http://vb.twbbs.org>

### Problem 1：字串問題

子題 1：計算字數。(程式執行限制時間: 2 秒) **9 分**

做網路廣告的時候，有些廣告的文案都有規定字數不能超過多少，如果想要知道每列測試資料(每組測試資料)有多少英文字，字和字之間用一個或多個空白隔開。 ” , ; ! . ” 這四個符號會與英文字相鄰。寫一程式計算每列字數。

#### 輸入說明：

第 1 列的數字  $n$  代表有幾組資料要測試， $2 \leq n \leq 8$ ，第二列起為測試資料，每列為一組測試資料，每組測試資料字元數  $\leq 255$ 。

#### 輸出說明：

每組測試資料輸出一列，計算每列字數。

#### 輸入檔案 1：【檔名：in1.txt】

3

This is a sample file.

Hello World!!

Hi!

#### 輸入檔案 2：【檔名：in2.txt】

3

Bigtable timestamps are integers.

Each cell in a Bigtable can contain multiple versions of the same data; these versions are indexed by timestamp.

It stood on a hill overlooking the village, some of its windows boarded, tiles missing from its roof, and ivy spread unchecked over its face.

#### 輸出範例：【檔名：out.txt】

5

2

1

4

19

25

**子題 2：摩斯電碼。** (程式執行限制時間: 2 秒) 11 分

摩斯電碼是一種在電報通訊的時代傳送文字訊息的方式，透過不同的排列順序來表達不同的英文字母、數字和標點符號。是由美國人薩繆爾·摩斯在 1836 年發明。有兩種「符號」用來表示字元：點 (.) (ASC 碼 0X2E) 和劃 (-) (ASC 碼 0X2D)，或叫「滴」(Dit) 和「答」(Dah)。數字對應的摩斯電碼如下：

0	----- (Dah-Dah-Dah-Dah-Dah)
1	.---- (Dit-Dah-Dah-Dah-Dah)
2	..--- (Dit-Dit-Dah-Dah-Dah)
3	...-- (Dit-Dit-Dit-Dah-Dah)
4	....- (Dit-Dit-Dit-Dit-Dah)
5	..... (Dit-Dit-Dit-Dit-Dit)
6	-.... (Dah-Dit-Dit-Dit-Dit)
7	--... (Dah-Dah-Dit-Dit-Dit)
8	---.. (Dah-Dah-Dah-Dit-Dit)
9	----. (Dah-Dah-Dah-Dah-Dit)

寫一程式，把摩斯電碼轉成摩斯電碼對應到的數字。

**輸入說明：**

第 1 列的數字  $n$  代表有幾筆資料要測試， $2 \leq n \leq 8$ ，第二列起為測試資料，之後每列為每筆的測試資料，即是要解譯的摩斯電碼，每列摩斯電碼字元中間用一個或多個空白隔開。每列摩斯電碼對應到的數字(字元數)  $\leq 10$ 。

**輸出說明：**

每筆測試資料輸出一列，為摩斯電碼對應到的數字。

輸入檔案 1 :【檔名：in1.txt】

3

.--- .--- ...-  
....- .... -....  
-... -... -....

輸入檔案 2 :【檔名：in2.txt】

2

.--- .-----  
----- .--- .--- ...- ....- .... -.... -... -... -....

輸出範例 :【檔名：out.txt】

123

456

789

110

0123456789

## Problem 2：數學問題

### 子題 1：網段 ID (程式執行限制時間: 2 秒) 13 分

IPv4 的 IP 位址為 32 位元資料，內容常表示成 4 個十進位數字，中間以點(.)隔開(如 192.168.10.246)

那要如何從一個 IP 位址和子網路遮罩得知網段 ID 呢？

實際的情形是：當知道 IP 位址 和 子網路遮罩 (都是二進位數字)之後，可使用一個 AND 的二進位邏輯運算，來求出網段 ID。我們任選一個 IP 位址來做例子：

IP 位址：139.175.153.252 換成二進位是：

10001011.10101111.10011001.11111100

給予子網路遮罩是 255.255.0.0，換成二進位：

11111111.11111111.00000000.00000000

然後將 IP 位址和子網路遮罩以AND運算：

10001011.10101111.10011001.11111100

AND

11111111.11111111.00000000.00000000

得出：

10001011.10101111.00000000.00000000

換成十進位就是 139.175.0.0，這個就是網段 ID 了。

範例：

測試資料會有 IP 位址/子網路遮罩資訊，寫一程式計算網段 ID。

IP 位址	192.168.10.65	11000000.10101000.00001010.010 00001
子網路遮罩	255.255.255.224	11111111.11111111.11111111.111 00000
網段 ID	192.168.10.64	11000000.10101000.00001010.010 00000

IP 位址	10.240.168.19	00001010.11110000.10 101000.00010011
子網路遮罩	255.255.192.0	11111111.11111111.11 000000.00000000
網段 ID	10.240.128.0	00001010.11110000.10 000000.00000000

**輸入說明：**

第一列的數字  $n$  代表有幾筆資料要測試， $2 \leq n \leq 8$ ，之後每列為每筆的測試資料，共有二個資料，內容為 IP 位址和子網路遮罩，中間以“/”隔開。

例如 139.175.153.252/255.255.0.0 這組測試資料中，139.175.153.252 為 IP 位址，255.255.0.0 為子網路遮罩。

**輸出說明：**

每筆測試資料輸出一列。依 IP 位址/子網路遮罩資訊，輸出網段 ID。

**輸入檔案 1：【檔名：in1.txt】**

3

139.175.153.252/255.255.0.0

192.168.10.65/255.255.255.224

10.240.168.19/255.255.192.0

**輸入檔案 2：【檔名：in2.txt】**

2

172.16.115.65/255.255.254.0

192.168.168.19/255.255.255.240

**輸出範例：【檔名：out.txt】**

139.175.0.0

192.168.10.64

10.240.128.0

172.16.114.0

192.168.168.16

**子題 2：最大公約數計算。**(程式執行限制時間: 2 秒) **12 分**

最大公因數 (Greatest Common Divisor, 簡寫為 GCD)，算出二個正整數共有因數。例如數字 20,8 的最大公因數(GCD)為 4。

$$20 = 2^2 \times 5^1$$

$$8 = 2^3$$

$$\text{GCD}(20,8)=4$$

給你  $M$  個正整數，任二個正整數，找出他們所有之中最大的一對 GCD 值。

**輸入說明：**

第一列的數字  $n$  代表有幾筆資料要測試， $2 \leq n \leq 8$ ，第二列起為測試資料，之後每列為每組測試資料，每組測試資料至少有 2 個正整數最多有 5 個正整數，正整數數字  $N$ ， $2 \leq N \leq 65535$ 。各個數字間以“,”隔開。每組資料都有  $M$  個正整數 ( $2 \leq M \leq 5$ )，請找其中的最大的一對 GCD 值。

**輸出說明：**

每組測試資料輸出一列。對於每組資料( $M$  個正整數)，請輸出最大的一對 GCD 值。

**輸入檔案 1：【檔名：in1.txt】**

3

10,20,30,40

125,15,25

8,12

**輸入檔案 2：【檔名：in2.txt】**

3

24,60,36

2,65535

2,3,4,5,6

**輸出範例：【檔名：out.txt】**

20

25

4

12

1

3

### Problem 3：資料結構—樹

子題 1：是否為堆積樹(Heap tree)或二元搜尋樹(Binary search tree)。(程式執行限制時間: 2 秒)  
14 分

在資料結構中，樹狀結構是可以用來描述有分支的結構，包含 1 個或多個節點。其存在一個特殊的節點，稱為根節點(root)，可連結若干子樹，也可以沒有子樹；從任一節點到根節點，都只有唯一的節點不重複路徑。

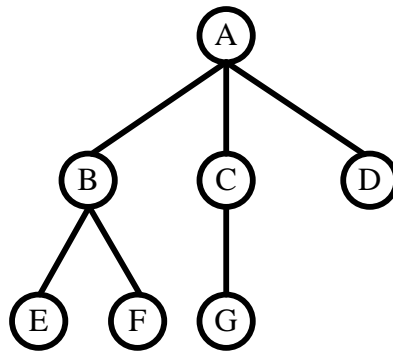


圖 3.1.1

在圖 3.1.1 中，有編號的圓形代表節點，A 為根節點，B、C 及 D 均為 A 的子節點，各節點之間不會有迴圈，且所有節點之間都有一個或多個邊相連通。任一樹狀結構的總邊數等於其總節點數減 1，在樹上任意添加一條邊，就會產生迴圈。

專有名詞介紹：

- (1) 無父節點的節點為根節點(Root)，如 A。
- (2) 父節點 (Parent)：一節點的上層節點為父節點，如 B 的父節點為 A，如 G 的父節點為 C。
- (3) 子節點 (Children)：一節點的下層節點為子節點，如 B 的子節點有 E 及 F；C 的子節點有 G。
- (4) 兄弟節點 (Siblings)：有共同父節點的節點稱為兄弟節點，如 B、C、D 互為兄弟節點。
- (5) 分支度 (Degree)：一個節點的子樹個數稱為其分支度，如 A 的分支度為 3；B 的分支度為 2；C 的分支度為 1；E 的分支度為 0。
- (6) 樹葉節點(Terminal node)：無子節點的節點，如 D、E、F、G。
- (7) 內部節點 (Non-terminal node)：樹葉以外的節點均為內部節點，如 A、B、C。
- (8) 階層或階度 (Level)：A 為階層 1；B、C、D 為階層 2；E、F、G 為階層 3。
- (9) 高度 (Height)：樹的最大階度，例如圖 3.1.1，因最大階度階度為 3，則其樹的高度為 3。

堆積樹(Heap tree)是一個二元樹，每個父節點最多只有兩個子節點，堆積樹的父節點若小於子節點，則稱之為最小堆積 (Min heap tree)，父節點若大於子節點，則稱之為最大堆積 (Max heap tree)，而同一層的子節點則無需理會其大小關係。

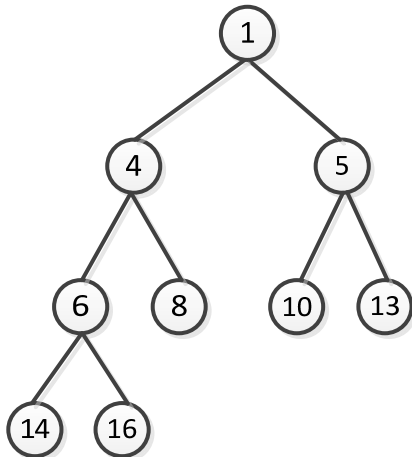


### 最小堆積樹(Min heap tree)

指每一個節點的鍵值必須小於它的子節點的鍵值。其特性如下：

1. 每一棵 Min heap tree 是一棵「完整二元樹」(Complete Binary Tree)。
2. 樹根的鍵值小於左子樹與右子樹的鍵值。
3. 其左子樹與右子樹亦是 Min heap tree。如下圖所示：

將下圖的堆積樹轉換為一維陣列之後如下所示：{1,4,5,6,8,10,13,14,16}

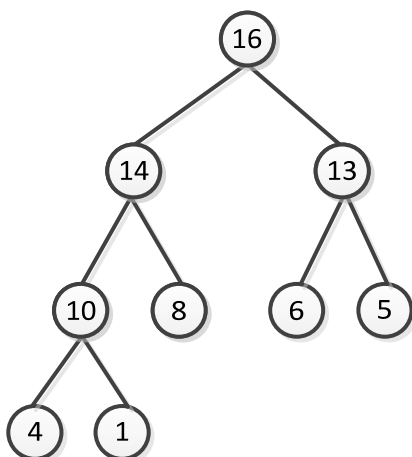


### 最大堆積樹(Max heap tree)

指每一個節點的鍵值必須大於它的子節點的鍵值。其特性如下：

1. 每一棵 Max heap tree 是一棵「完整二元樹」(Complete Binary Tree)。
2. 樹根的鍵值大於左子樹與右子樹的鍵值。
3. 其左子樹與右子樹亦是 Max heap tree。如下圖所示：

將下圖的堆積樹轉換為一維陣列之後如下所示：{16,14,13,10,8,6,5,4,1}



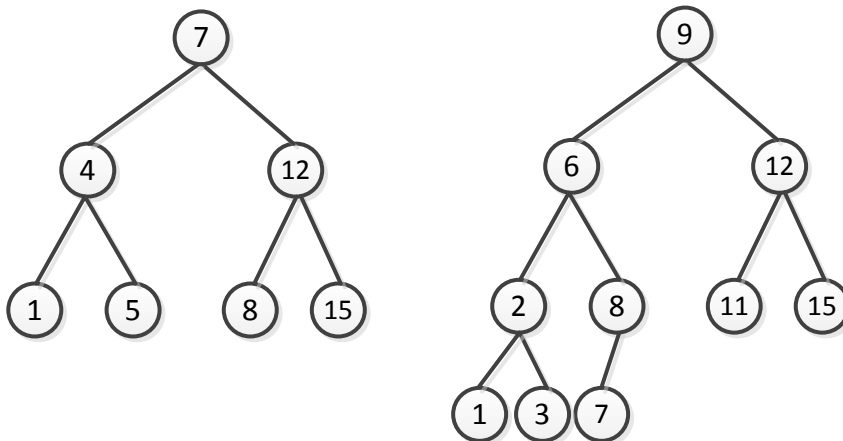
而在一棵二元樹中，除最後一層外，若其餘層都是滿的，並且最後一層或者是滿的，或者是在右邊缺少連續若干節點，則此二元樹為「完整二元樹」(Complete Binary Tree)。

二元搜尋樹(Binary search tree)定義:

二元搜尋樹是一種二元樹，它可以為空，若不為空，則必須要滿足以下條件：

- 1.若左子樹不為空，則左子樹的鍵值均須要小於樹根的鍵值。
- 2.若右子樹不為空，則右子樹的鍵值均須要大於樹根的鍵值。
- 3.左子樹與右子樹必須也要保持二元搜尋樹。

將下圖的二元搜尋樹(「完整二元樹」(Complete Binary Tree))轉換為一維陣列之後如下所示：  
{7,4,12,1,5,8,15}和{9,6,12,2,8,11,15,1,3,7}。



寫一個程式，讀入一資料，資料內容為「完整二元樹」(Complete Binary Tree)，然後回答該資料是否為堆積樹(Heap tree)或二元搜尋樹(Binary search tree)。

#### 輸入說明：

第一列的數字  $n$  代表共有幾組資料要測試， $2 \leq n \leq 8$ 。

第二列起每一列代表一組測試資料。每組測試資料代表一「完整二元樹」(Complete Binary Tree)。測試資料為多個不同的數字  $x_i$ ， $0 \leq x_i \leq 65535$ ， $4 \leq |x_i| \leq 64$ ，中間用逗號隔開。

#### 輸出說明：

每組測試資料輸出一列。輸出每組測試資料是否為堆積樹(Heap tree)或二元搜尋樹(Binary search tree)。若該資料是堆積樹(Heap tree)，則輸出 H；若該資料是二元搜尋樹(Binary search tree)，則輸出 B；若不是堆積樹(Heap tree)同時也不是二元搜尋樹(Binary search tree)，則輸出 F。不會有同時為堆積樹(Heap tree)和二元搜尋樹(Binary search tree)的測試資料。

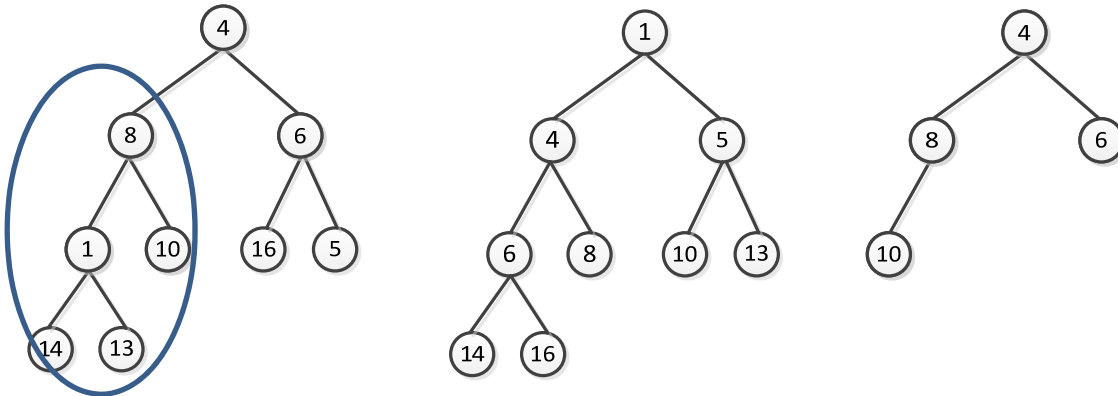
輸入檔案 1 :【檔名：in1.txt】

3

4,8,6,1,10,16,5,14,13

1,4,5,6,8,10,13,14,16

4,8,6,10



輸入檔案 2 :【檔名：in2.txt】

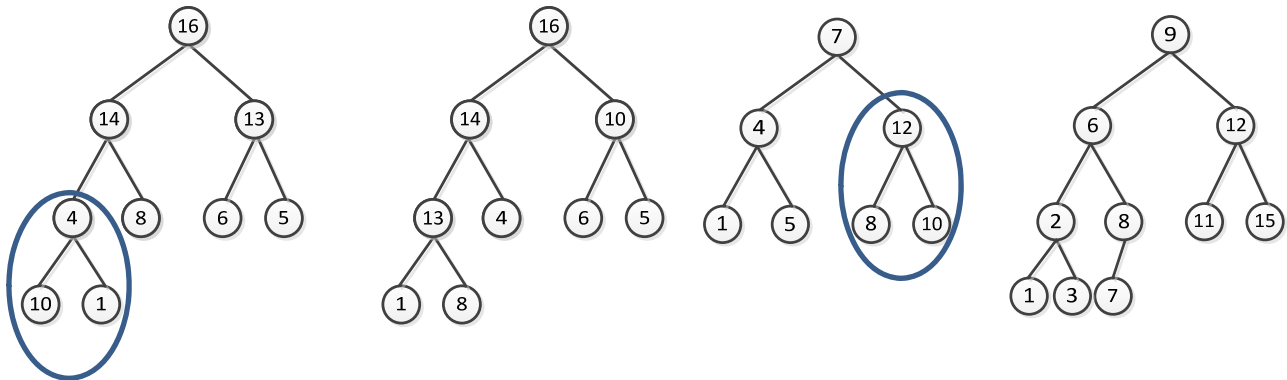
4

16,14,13,4,8,6,5,10,1

16,14,10,13,4,6,5,1,8

7,4,12,1,5,8,10

9,6,12,2,8,11,15,1,3,7



輸出範例 :【檔名：out.txt】

F

H

H

F

H

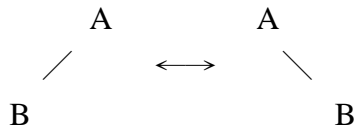
F

B

## 子題 2：後序表示法 (post-order)。(程式執行限制時間: 2 秒) 15 分

二元樹的定義：

1. 樹不可以為空集合，亦即至少必須有一個根節點，但二元樹卻可以是空集合。
2. 樹的兄弟節點位置次序並非固定，但二元樹是固定的。也就是下面是相同的樹，但卻不是相同的二元樹。



在二元樹的運用上，常常需要找出所有的節點資料，這個過程稱為樹的拜訪或追蹤。依拜訪追蹤的次序可分成下列三種：前序表示法 (pre-order)、中序表示法 (in-order) 及後序表示法 (post-order)。

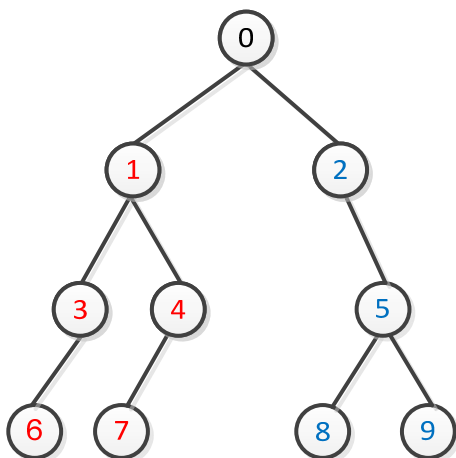
### 二元樹的走訪 (Traversal of Tree)

對於一個二元樹，我們有三種最常用的方法可以走過這棵樹所有的節點。

1. 前序表示法 (pre-order)：根節點 -> 左子樹 -> 右子樹
2. 中序表示法 (in-order)：左子樹 -> 根節點 -> 右子樹
3. 後序表示法 (post-order)：左子樹 -> 右子樹 -> 根節點

有兩種序，就有機會還原出唯一的一棵二元樹。比方說，知道中序表示法 (in-order) 和前序表示法 (pre-order)，可以求出原本的二元樹。

在前序表示法 (pre-order) 之中，最左邊的元素就是 root；在中序表示法 (in-order) 之中，root 的兩邊分別為左子樹和右子樹 —— 利用 root 便可區分左子樹和右子樹。子樹也是樹，可以用相同手法繼續分割，最後便可求出整棵二元樹的架構。



中序表示法 (in-order) : 6,3,1,7,4,0,2,8,5,9

前序表示法 (pre-order) : 0,1,3,6,4,7,2,5,8,9

### 輸入說明：

第一列的數字  $n$  代表有幾組資料要測試， $2 \leq n \leq 8$ ，第二列起為每組的測試資料，之後每二列為每組的測試資料。每組測試資料的第一列為中序表示法（in-order）；每組測試資料的第二列為前序表示法（pre-order），各節點編號不會相同。測試資料為多個數字  $x_i$ ， $0 \leq x_i \leq 65535$ ， $4 \leq |x_i| \leq 64$ ，中間用逗號隔開。用測試資料找出整棵二元樹的架構。

### 輸出說明：

在測試資料中所建二元樹，輸出這棵二元樹之後序表示法（post-order），每組測試資料輸出一列。

#### 輸入檔案 1：【檔名：in1.txt】

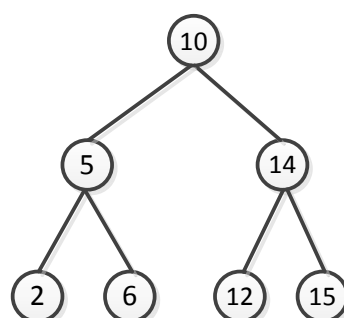
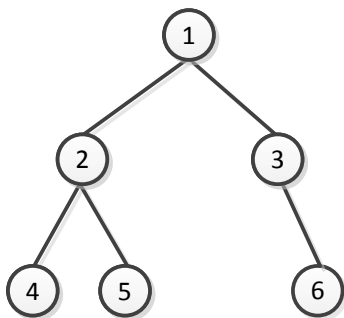
2

4,2,5,1,3,6

1,2,4,5,3,6

2,5,6,10,12,14,15

10,5,2,6,14,12,15



#### 輸入檔案 2：【檔名：in2.txt】

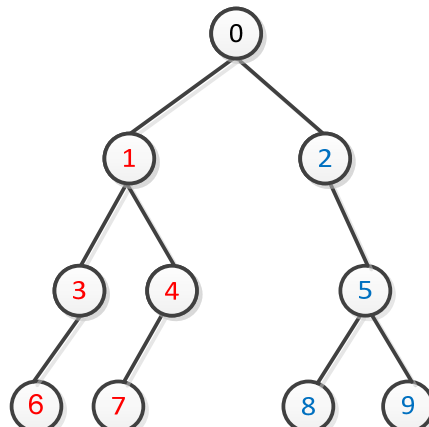
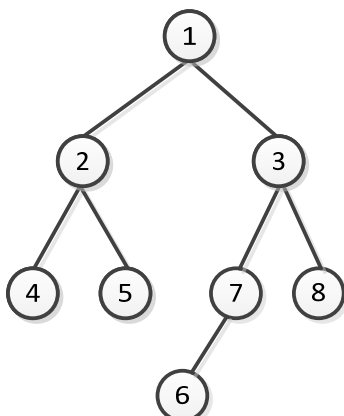
2

4,2,5,1,6,7,3,8

1,2,4,5,3,7,6,8

6,3,1,7,4,0,2,8,5,9

0,1,3,6,4,7,2,5,8,9

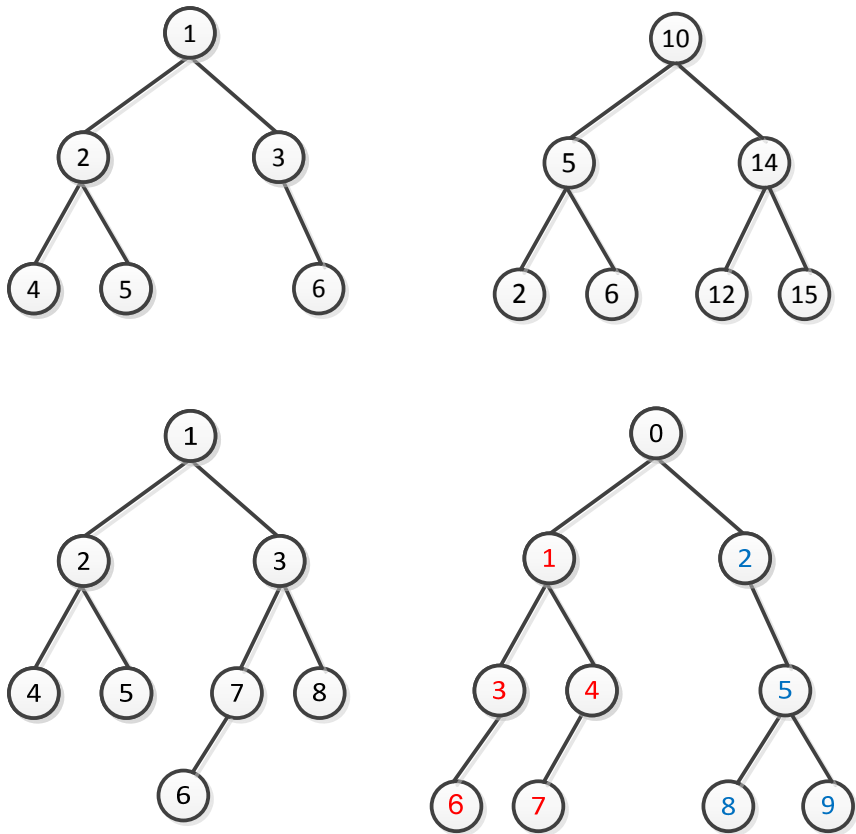


輸出範例：【檔名：out.txt】

4,5,2,6,3,1  
2,6,5,12,15,14,10

4,5,2,6,7,8,3,1  
6,3,7,4,1,8,9,5,2,0

方便選手比對結果，把上一頁的圖，這也放相同的一份。



#### Problem 4：其它

子題 1：最長共同子序列(Longest common subsequence)。(程式執行限制時間: 2 秒) 10 分

「最長共同子序列」 (Longest Common Subsequence, LCS)。LCS 是兩個序列(sequence)，在各自所有的子序列(subsequence)之中，一模一樣而且最長的那個子序列(subsequence)。

給 2 個字串，請你輸出他們的最長共同子序列 (Longest common subsequence) 的長度。

也就是說，在這兩個字串各自所有的子序列之中，內容相同而且長度最長的那個子序列。舉例來說有兩個字串 abcdghxy 和 aedfhrz，它們的最長共同子序列為 adh，長度為 3。

j		0	1	2	3	4	5	6	7
i			a	e	d	f	h	r	z
0	X	0	0	0	0	0	0	0	0
1	a	0	1	1	1	1	1	1	1
2	b	0	1	1	1	1	1	1	1
3	c	0	1	1	1	1	1	1	1
4	d	0	1	1	2	2	2	2	2
5	g	0	1	1	2	2	2	2	2
6	h	0	1	1	2	2	3	3	3
7	x	0	1	1	2	2	3	3	3
8	y	0	1	1	2	2	3	3	3

<http://lcs-demo.sourceforge.net/>

定義：給定兩個序列(Sequence) $X = \langle x_1, x_2, \dots, x_m \rangle$ 和 $Y = \langle y_1, y_2, \dots, y_n \rangle$ ,求 X 和 Y 所構成的最長共同子序列  $LCS(X,Y) = Z = \langle z_1, z_2, \dots, z_k \rangle$ 為何？

[注意] 子字串(Substrings)和子序列(Subsequences)的差別

例如有一條字串(String)  $S = \text{"a t g a t g c a a t"}$ ，則

子字串(Substrings) of S : "g a t g c", "t g c a a t"

子序列(Subsequences) of S : "a g g t", "a a a a"

解法

$$C[i,j] = \begin{cases} 0, & \text{if } i = 0 \text{ or } j = 0 \\ C[i-1, j-1] + 1, & \text{if } i, j > 0 \text{ and } x_i = y_j \\ \text{Max}(C[i, j-1], C[i-1, j]), & \text{if } i, j > 0 \text{ and } x_i \neq y_j \end{cases}$$

若 $i$ 或 $j$ 為0，表示 $X$ 或 $Y$ 這兩條序列的其中一條為空序列。

若 $x_i = y_j$ ，則 $c[i, j]$ 所表示的序列長度，是由 $\langle x_1, x_2, \dots, x_{i-1} \rangle$ 和 $\langle y_1, y_2, \dots, y_{j-1} \rangle$ 兩序列所構成之最長共同子序列的長度，即 $c[i-1, j-1]$ 再加上1。

若 $x_i \neq y_j$ ，則 $c[i, j]$ 的序列長度是由下列兩個不同的最長共同子序列長度當中之最大值所構成：

1.  $\langle x_1, x_2, \dots, x_{i-1} \rangle$ 和 $\langle y_1, y_2, \dots, y_j \rangle$ 兩序列所構成之最長共同子序列的長度 $c[i-1, j]$

2.  $\langle x_1, x_2, \dots, x_i \rangle$ 和 $\langle y_1, y_2, \dots, y_{j-1} \rangle$ 兩序列所構成之最長共同子序列的長度 $c[i, j-1]$

```
function LCSLength(X[1..m], Y[1..n])
    C = array(0..m, 0..n)
    for i := 0..m
        C[i,0] = 0
    for j := 0..n
        C[0,j] = 0
    for i := 1..m
        for j := 1..n
            if X[i] = Y[j]
                C[i,j] := C[i-1,j-1] + 1
            else
                C[i,j] := max(C[i,j-1], C[i-1,j])
    return C[m,n]
```

[https://en.wikipedia.org/wiki/Longest\\_common\\_subsequence\\_problem](https://en.wikipedia.org/wiki/Longest_common_subsequence_problem)

**輸入說明：**

第一列的數字  $n$  代表有幾組資料要測試， $2 \leq n \leq 8$ ，第二列起為每組的測試資料，之後每二列為每組的測試資料。每組測試資料為二列字串，字串內容為英文字母或數字，大小寫的英文字母不同，每列最多有 80 個字元。

**輸出說明：**

對輸入的每組測試資料，輸出它們最長共同子序列的長度。



輸入檔案 1 :【檔名：in1.txt】

2

ACCGATGCAGCGCTC

CCGATGA

abcdghxy

aedfhrz

輸入檔案 2 :【檔名：in2.txt】

2

13567

24680

ab12

Abc1

輸出範例 :【檔名：out.txt】

7

3

**1**

2

## 子題 2：霍夫曼編碼 (Huffman Coding)。(程式執行限制時間: 2 秒) 16 分

霍夫曼編碼 (Huffman Coding)，又譯為哈夫曼編碼、赫夫曼編碼，是一種用於無損資料壓縮演算法。由大衛·霍夫曼在 1952 年發明。

在電腦資料處理中，霍夫曼編碼使用變長編碼表對源符號 (如檔案中的一個字母) 進行編碼。例如，在英文中，e 的出現機率最高，而 z 的出現機率則最低。當利用霍夫曼編碼對一篇英文進行壓縮時，e 極有可能用一個位元來表示，而 z 則可能花去 25 個位元 (不是 26)。用普通的表示方法時，每個英文字母均占用一個字元組，即 8 個位元。二者相比，e 使用了一般編碼的 1/8 的長度，z 則使用了 3 倍多。倘若我們能對於英文中各個字母出現機率較準確的估算，就可以大幅度提高無失真壓縮的比例。

霍夫曼編碼 (Huffman Coding) 要先建立霍夫曼樹(Huffman Tree):

建立霍夫曼樹(Huffman Tree)步驟：

(一)針對相異字元，統計其出現的次數：

字元	A	B	C	D	E	F	G
出現的次數	2	3	6	8	13	15	19

(二)為每個字元建立一顆只有一節點的樹，每棵樹的根節點之關鍵值(紅色字)為其字元出現的次數.

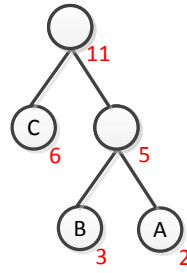
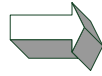
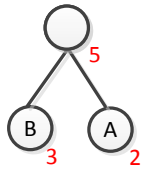


(三)找出根節點關鍵值(出現次數)最小的兩顆樹。

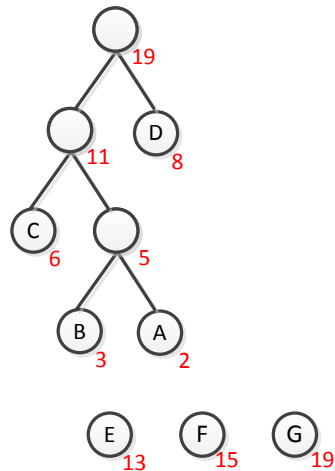
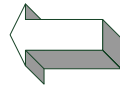
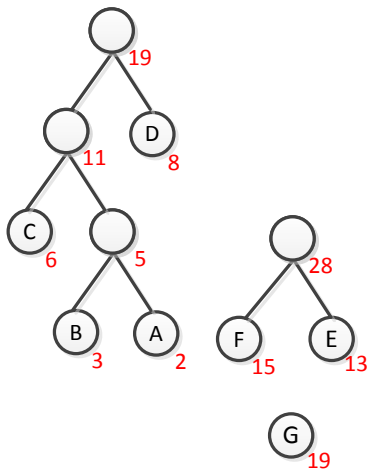
(四)產生一個新的根節點，並將找到的兩棵樹分別當作此新的根節點之左右子樹(節點關鍵值大的放左樹，關鍵值小的放右樹或是節點關鍵值小的放左樹，關鍵值大的放右樹)，而根節點的關鍵值為左右子樹節點之關鍵值(紅色字)的和.

(五)重複步驟 (三) 與 (四)，直至全部節點合併為一棵樹。

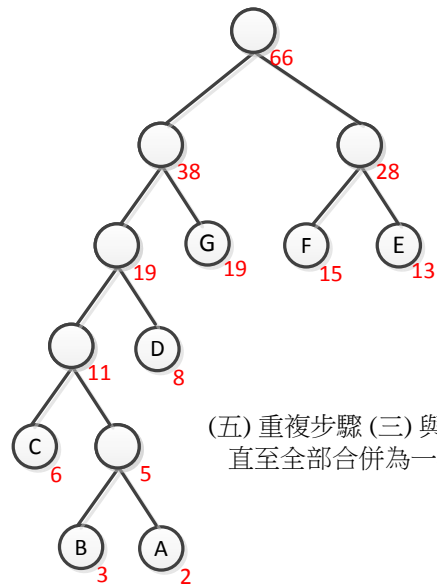
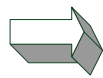
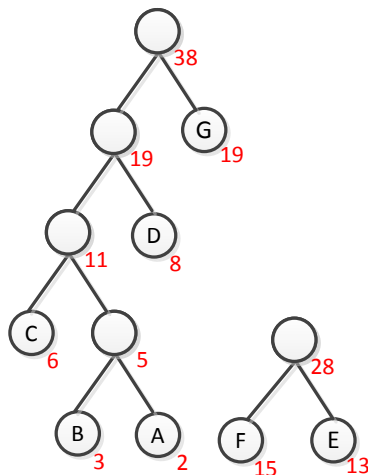
(四) 產生一個根節點，節點關鍵值大的放左樹，關鍵值小的放右樹，而根節點的關鍵值為左右子樹節點之關鍵值(紅色字)的和 $2+3=5$ 。



(三) 找出根節點關鍵值(出現次數)最小的兩顆樹2,3。



(三) 找出根節點關鍵值(出現次數)最小的兩顆樹13,15。(三) 找出根節點關鍵值(出現次數)最小的兩顆樹8,11。



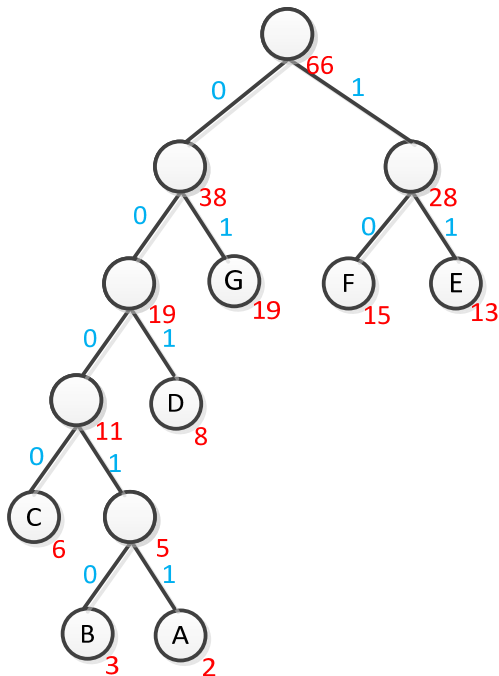
(五) 重複步驟(三)與(四)，直至全部合併為一棵樹。

(三) 找出根節點關鍵值(出現次數)最小的兩顆樹19,19。二個關鍵值一樣，可以安照排序先後秩序。(三) 找出根節點關鍵值(出現次數)最小的兩顆樹28,38。

產生霍夫曼編碼(Huffman Code) :

(一)在霍夫曼樹(Huffman Tree) 中，針對每個節點，將連至左子樹的邊標為 0，將連至右子樹的邊標示為 1。

\* 注意一：霍夫曼樹(Huffman Tree) 的每個葉節點代表一個相異字元，且葉節點的個數恰等於相異字元的個數。



(二)針對每個由根節點至葉節點的路徑，將其所經過邊的標示連結起來，並指派給對應葉節點所代表的字元，此即霍夫曼編碼(Huffman Code)：

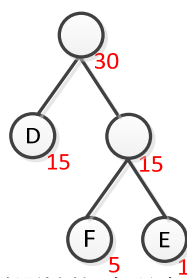
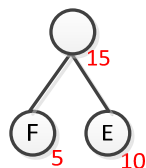
字元	A	B	C	D	E	F	G
出現的次數	2	3	6	8	13	15	19
霍夫曼編碼	00011	00010	0000	001	11	10	01
霍夫曼編碼 位元數(長度)	5	5	4	3	2	2	2

字元	A	B	C	D	E	F
出現的次數	26	25	20	15	10	5
霍夫曼編碼	10	01	00	110	1111	1110
霍夫曼編碼 位元數(長度)	2	2	2	3	4	4

壓縮檔案：

當產生所有字元的霍夫曼編碼(Huffman Code)後，我們可以利用霍夫曼編碼(Huffman Code)來取代檔案中的所有字元。

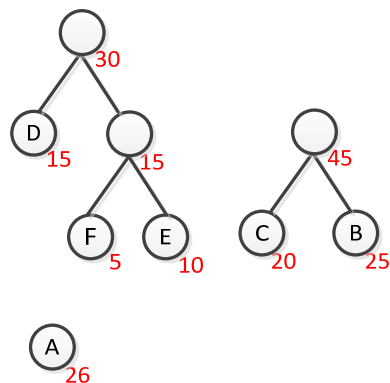
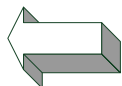
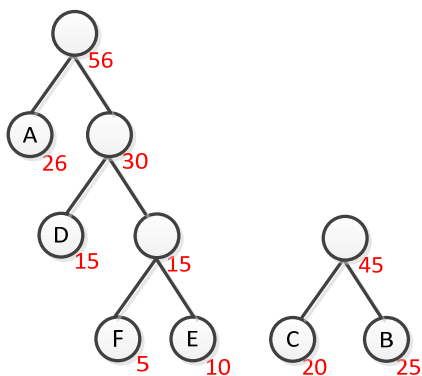
(四) 產生一個根節點，節點關鍵值小的放左樹，關鍵值大的放右樹，而根節點的關鍵值為左右子樹節點之關鍵值(紅色字)的和 $5+10=15$ 。



(三) 找出根節點關鍵值(出現次數)最小的兩顆樹5,10。

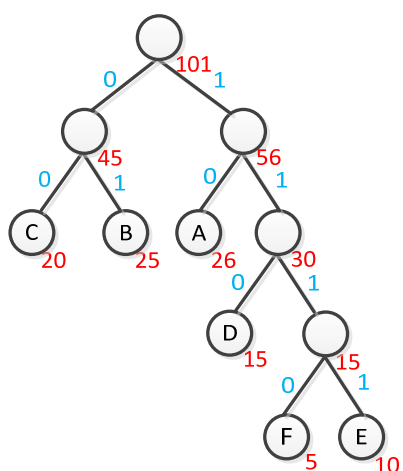
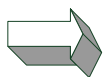
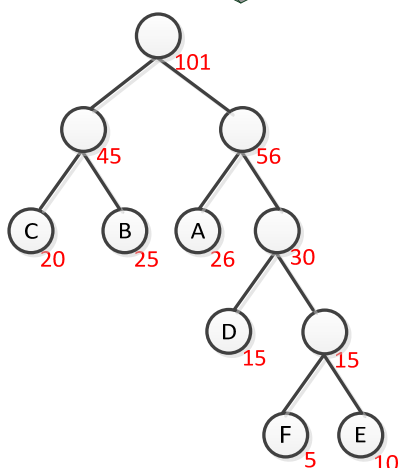


(三) 找出根節點關鍵值(出現次數)最小的兩顆樹15,15。二個關鍵值一樣，可以安照排序先後秩序。左右順序任意放不會影響答案霍夫曼編碼位元數(長度)。



(三) 找出根節點關鍵值(出現次數)最小的兩顆樹26,30。

(三) 找出根節點關鍵值(出現次數)最小的兩顆樹20,25。



(三) 找出根節點關鍵值(出現次數)最小的兩顆樹45,56。

產生霍夫曼編碼(Huffman Code)

**輸入說明：**

第一列的數字  $n$  代表有幾筆資料要測試， $2 \leq n \leq 8$ ，第二列起為測試資料，之後每列為每組測試資料，每組測試資料至少有 2 個正整數最多有 26 個正整數，正整數數字  $N$ ， $1 \leq N \leq 65535$ 。各個數字間以“,”隔開，分別代表各字元的出現的次數。每組資料都有  $M$  個相異數字 ( $2 \leq M \leq 26$ )，讓你去建立霍夫曼樹(Huffman Tree)。會避免 3,3,8,8 這樣輸出結果不唯一的測試資料。

**輸出說明：**

在測試資料中所建建立霍夫曼樹(Huffman Tree)，完成霍夫曼編碼(Huffman Code)及計算霍夫曼編碼(Huffman Code)位元數(長度)，這組測試資料輸出一列，各個數字間以“,”隔開，分別代表各字元的霍夫曼編碼(Huffman Code)位元數(長度)。

**輸入檔案 1：【檔名：in1.txt】**

```
2
2,3,6,8,13,15,19
4,8,5
```

**輸入檔案 2：【檔名：in2.txt】**

```
3
26,25,20,15,10,5
4,3,5,6
2,1
```

**輸出範例：【檔名：out.txt】**

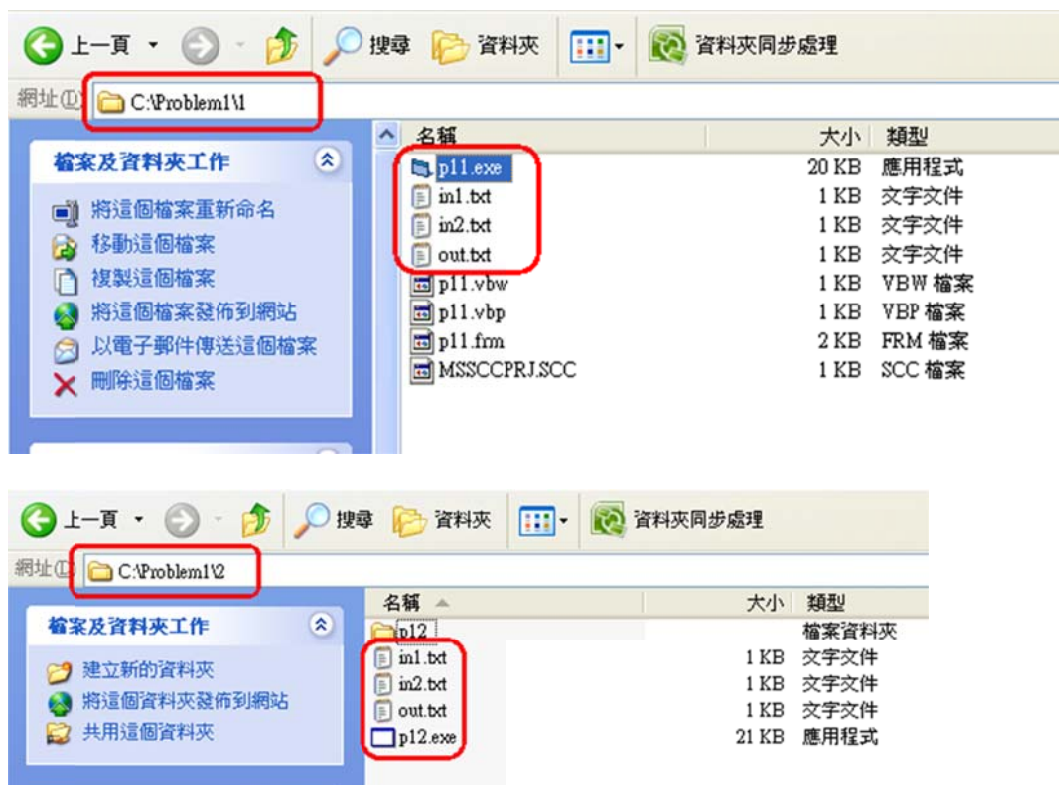
```
5,5,4,3,2,2,2
2,1,2

2,2,2,3,4,4
2,2,2,2
1,1
```

選手在作答前，先將主辦單位發放的”光碟”內資料，全部複製到選手使用的電腦硬碟 C 槽中。光碟內容包括 4 個資料夾(「Problem1」到「Problem4」)，各資料夾內有「1」及「2」兩個子資料夾。在子資料夾中，已存有該子題的輸入資料檔。選手在競賽時，各子題的程式及輸出檔，應產生在硬碟對應的子資料夾中。考試結束時，在硬碟 C 槽的考試資料夾(含子題的整個專案、程式、輸入及輸出檔)，請選手全部再「備份」回原隨身碟中。

1. 各個子題均提供 2 組測試輸入檔，檔名分別是「in1.txt」及「in2.txt」。選手製作的程式，應依序讀入「in1.txt」及「in2.txt」檔，並產生 1 個輸出檔「out.txt」。(即，每個程式讀入 2 個輸入檔，產生 1 個輸出檔。)在輸出檔中，選手應先輸出「in1.txt」產生的結果，再輸出「in2.txt」的結果，兩組結果間用 1 行「空白行」隔開。不影響結果的空白鍵，不列入扣分。若程式執行檔執行結果未依序、不全或無法執行，該子題以零分計算。
2. 程式原始檔的「專案名稱」請依規定命名，程式中使用的「表單」名稱請與專案名稱命名相同。若選手在同一專案中使用多個表單，依開發環境內定名稱命名即可。輸入資料檔、輸出資料檔、程式執行檔請依”存放路徑及檔名”規定存放。若選手使用有別於 VB 之開發環境，專案(程式)名稱仍依規定。

各個子題均提供 2 組測試輸入檔，檔名分別是「in1.txt」及「in2.txt」。選手製作的程式，應將「in1.txt」「in2.txt」「out.txt」及 p??.exe(例如：p11.exe 或 p12.exe)，這四個檔案放在指定的地方。如下圖所示：



題目	子題	檔案類型	存放路徑及檔名
<b>Problem1</b>	<b>子題 1</b>	輸入資料檔	C:\Problem1\1\in1.txt 和 in2.txt
		輸出資料檔	C:\Problem1\1\out.txt
		程式執行檔	C:\Problem1\1\p11.exe
	<b>子題 2</b>	輸入資料檔	C:\Problem1\2\in1.txt 和 in2.txt
		輸出資料檔	C:\Problem1\2\out.txt
		程式執行檔	C:\Problem1\2\p12.exe

題目	子題	檔案類型	存放路徑及檔名
<b>Problem2</b>	<b>子題 1</b>	輸入資料檔	C:\Problem2\1\in1.txt 和 in2.txt
		輸出資料檔	C:\Problem2\1\out.txt
		程式執行檔	C:\Problem2\1\p21.exe
	<b>子題 2</b>	輸入資料檔	C:\Problem2\2\in1.txt 和 in2.txt
		輸出資料檔	C:\Problem2\2\out.txt
		程式執行檔	C:\Problem2\2\p22.exe

題目	子題	檔案類型	存放路徑及檔名
<b>Problem3</b>	<b>子題 1</b>	輸入資料檔	C:\Problem3\1\in1.txt 和 in2.txt
		輸出資料檔	C:\Problem3\1\out.txt
		程式執行檔	C:\Problem3\1\p31.exe
	<b>子題 2</b>	輸入資料檔	C:\Problem3\2\in1.txt 和 in2.txt
		輸出資料檔	C:\Problem3\2\out.txt
		程式執行檔	C:\Problem3\2\p32.exe

題目	子題	檔案類型	存放路徑及檔名
<b>Problem4</b>	<b>子題 1</b>	輸入資料檔	C:\Problem4\1\in1.txt 和 in2.txt
		輸出資料檔	C:\Problem4\1\out.txt
		程式執行檔	C:\Problem4\1\p41.exe
	<b>子題 2</b>	輸入資料檔	C:\Problem4\2\in1.txt 和 in2.txt
		輸出資料檔	C:\Problem4\2\out.txt
		程式執行檔	C:\Problem4\2\p42.exe