



# 全國高級中等學校 106 學年度商業類科學生技藝競賽

## 【 程式設計 】 職種【術科】模擬試卷

選手證號碼(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。

### Problem 1：字串問題

子題 1：計算字元數。(程式執行限制時間: 2 秒)

資料輸入的時候，有些文件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)有多字元數，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。

#### 輸入說明：

第 1 列的數字  $n$  代表有幾組資料要測試， $2 \leq n \leq 20$ ，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，計算每列字元數(含英文數字和符號)但不含空白。

#### 輸入檔案 1：【檔名：in1.txt】

3

Hi!

This is a sample file.

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

#### 輸入檔案 2：【檔名：in2.txt】

4

123 45 67 8 9 0 ab

123 45 67 8 9 0 ab ! @ #

123 45 67 8 9 ! @ #.

You shall have your sunset. I shall command it. But I shall wait, according to my science of government, until conditions are favorable.

輸出範例：【檔名：out.txt】

3

18

94

12

15

13

114

子題 2：前綴字串(Prefix) and 後綴字串(Postfix)。(程式執行限制時間: 2 秒)

字串 A 是字串 B 的前綴字串(Prefix)，若且唯若字串 B 的前  $\text{len}(A)$  個字母與 A 完全相同，其中  $\text{len}(A)$  指的是字串 A 的長度。例如：“exam”和“example”都是“example”的前綴字串(Prefix)，但是“mple”和“exap”都不是“example”的前綴字串(Prefix)。同樣的，當 B 的後  $\text{len}(A)$  個字母與 A 完全相同的時候，我們稱 A 是 B 的後綴字串(Postfix)，“mple”、“xample”和“example”都是“example”的後綴字串(Postfix)。給定兩個字串 P,Q，請你找出最長的字串 S，使得 S 是 P 的前綴字串(Prefix)，同時也是 Q 的後綴字串(Postfix)。

**輸入說明：**

第 1 列的數字  $n$  代表有幾筆資料要測試， $2 \leq n \leq 20$ ，第二列起，每二列為一組測試資料，兩個字串 P,Q 各佔一行，為小寫英文字母，長度皆不超過 100 字元。

**輸出說明：**

每組測試資料輸出一列。請你找出最長的字串 S，使得 S 是 P 的前綴字串(Prefix)，同時也是 Q 的後綴字串(Postfix)。輸出最長的字串 S 的長度  $\text{len}(S)$ 。

**輸入檔案 1：【檔名：in1.txt】**

```
4
example
exam
example
examp
example
mple
abcdef
xyzabcde
```

**輸入檔案 2：【檔名：in2.txt】**

```
2
abcdefg
fg
example
apple
```

輸出範例：【檔名：out.txt】

4

5

1

5

0

1

## Problem 2：身分證和羅馬數字符號問題

子題 1：身分證。(程式執行限制時間: 2 秒)

中華民國身分證的號碼是經由一串公式所產生出來的，其身分證字號共有十碼，包括第一個大寫的英文字母與接續的九個阿拉伯數字。

(1)第一個碼代表地區，轉換方式為：A 轉換成 1,0 兩個字元，B 轉換成 1,1，餘如下：

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
10	11	12	13	14	15	16	17	34	18	19	20	21	22	35	23	24	25	26	27	28	29	32	30	31	33

(2)第二個碼代表性別，1 代表男性，2 代表女性

(3)第三個碼到第九個字元為流水號碼。

(4)第十個碼為檢查號碼。

例如：A123456789，其 A 的轉換字元是 1 和 0，其餘各碼亦轉換成字元，依序存在  $n_1n_2n_3n_4n_5n_6n_7n_8n_9n_{10}n_{11}$  的變數中，如下：

1	0	1	2	3	4	5	6	7	8	9
$n_1$	$n_2$	$n_3$	$n_4$	$n_5$	$n_6$	$n_7$	$n_8$	$n_9$	$n_{10}$	$n_{11}$

然後再把每一個變數，依序乘上 1 9 8 7 6 5 4 3 2 1 及 1 的加權，再相加，如下：

$$n_1 \times 1 + n_2 \times 9 + n_3 \times 8 + n_4 \times 7 + n_5 \times 6 + n_6 \times 5 + n_7 \times 4 + n_8 \times 3 + n_9 \times 2 + n_{10} \times 1 + n_{11} \times 1$$

將身分證號碼 A123456789 套入公式，其結果為：

$$1 \times 1 + 0 \times 9 + 1 \times 8 + 2 \times 7 + 3 \times 6 + 4 \times 5 + 5 \times 4 + 6 \times 3 + 7 \times 2 + 8 \times 1 + 9 \times 1 \\ = 1 + 0 + 8 + 14 + 18 + 20 + 20 + 18 + 14 + 8 + 9 = 130$$

然後再除以 10，如果整除，該組身分證字號為有效。

假設今天有個活動，可以以有效的身分證字號進行報名。請寫一支程式判斷(1)有效的身分證字號 T 或(2)無效的身分證字號 F。

例如在測試資料中，有 6 筆資料：

1. M123456789(T)
2. A123456789(T)
3. A323456783(F)
4. A223344556(F)
5. R102345678(T)
6. A108881111(F)

### 輸入說明：

第一列的數字  $n$  代表有幾筆資料要測試， $2 \leq n \leq 20$ ，之後每列為每筆的測試資料，之後每列為每個測試身分證字號。在測試資料中 A323456783，套入公式，其身分證的驗證碼結果雖然正確，在第二個碼代表性別(1 代表男性，2 代表女性)，因其為 3，所以為無效的身分證字號。

### 輸出說明：

每行輸出 T,F，對應到每組測試資料中為有效的身分證字號 T 或無效的身分證字號 F。

輸入檔案 1：【檔名：in1.txt】

3

M123456789

A123456789

A323456783

輸入檔案 2：【檔名：in2.txt】

3

A223344556

R102345678

A108881111

輸出範例：【檔名：out.txt】

T

T

F

F

T

F

子題 2：給一個整數數字，轉為羅馬數字符號。(程式執行限制時間: 2 秒)

羅馬數字是古羅馬所使用的數字系統，以羅馬符號來表示數值。如果要對應成現在通用的阿拉伯數字，只用 7 個符號，包括 I(1)，V(5)，X(10)，L(50)，C(100)，D(500)，M(1000)。羅馬數字並沒有 0。羅馬數字 I 代表阿拉伯數字的 1，但 II 可不是代表 11，而是 1+1=2，III 為 3。如果要用羅馬數字來表示 4，不能寫成 IIII，而要寫成 IV，意思是 5-1=4。阿拉伯數字為 9 時，要寫成 IX。

(1)重複數次決定倍數：1 個羅馬數字重複幾次，就表示這個數的幾倍。例如: X X X=30。數碼限制：同樣數碼最多只能出現 3 次，如 40 不能表示為 XXXX，而要表示為 XL。

(2)右加左減：在 1 個較大的羅馬數字的右邊跟 1~3 個較小的羅馬羅馬數字，表示大數字加小數字。在一個較大的數字的左邊跟 1 個較小的羅馬數字，表示大數字減小數字。左減數字不能超過 1 位，比如 8 寫成 VIII，而非 IIX。右加數字最多只能 3 位，例如 14 寫成 XIV，而非 XIII。

(3)數字上加橫線乘千或乘百萬：在 1 個羅馬數字的上方加上 1 條橫線或者在右下方寫 M，表示將這個數字乘以 1000，即是原數的 1000 倍。同理，如果上方有 2 條橫線，即是原數的 1000000 倍。

羅馬數字與十進位數字的轉換範例：

IX = 9

VI = 6

XI = 11

XVII = 17

CCLXVIII=100 + 100 + 50 + 10 + 5 + 1 + 1 + 1 = 268

MMMCC=1,000 + 1,000 + 1,000 + 100 + 100 = 3200

DCCVII=500 + 100 + 100 + 5 + 1 + 1 = 707

MMCDLXIX= 1000 + 1000 + [500-100] + 50 + 10 + [10-1] = 2469

輸入說明：

第一列的數字  $n$  代表有幾筆資料要測試， $2 \leq n \leq 20$ ，之後每列為每筆的測試資料。

第二列之後就是每筆的測試資料為整數數字，這個整數數字會落在 1 到 3999 之間。

輸出說明：

將每筆的測試資料轉為羅馬數字符號。

**輸入檔案 1：【檔名：in1.txt】**

7  
1  
5  
10  
50  
100  
500  
1000

**輸入檔案 2：【檔名：in2.txt】**

8  
9  
6  
11  
17  
268  
3200  
707  
2469

**輸出範例：【檔名：out.txt】**

I  
V  
X  
L  
C  
D  
M  
  
IX  
VI  
XI  
XVII  
CCLXVIII  
MMMCC  
DCCVII  
MMCDLXIX



### Problem 3：數學問題

子題 1：解一元一次方程式。(程式執行限制時間: 2 秒)

題目用字串表示方程式，寫程式求出  $x$  的解(解一元一次方程式)，給定一個的方程並以字串“ $x = \# \text{ value}$ ”的形式求出  $x$  的值。方程式只包含“+”，“-”運算(不包含有 $*$ / $)$ ，變數  $x$  及其係數。如果方程有“無窮多的解” "Infinite solutions"，輸出 IS；如果方程式“無解” No solution，輸出 NS。

如果方程只有一個解，我們確保  $x$  的值是一個整數。解一元一次方程式，處理字串，將  $x$  的係數合併起來，將常數合併起來，化簡成  $ax=b$  的形式來求解。不會有  $2x=3$  這樣無法整除的測試資料。

範例：

(1) " $x+5-3+x=7+x-2$ "

➔ " $x=3$ "

(2) " $2x=x$ "

➔ " $x=0$ "

(3) " $2x+3x-6x=x+4$ "

➔ " $x=-2$ "

(4) " $2x=6$ "

➔ " $x=3$ "

(5) " $x=x$ "

➔ "無窮多的解"

(6) " $x=x+2$ "

➔ "無解"

輸入說明：

第一列的數字  $n$  代表有幾筆資料要測試， $2 \leq n \leq 20$ 。第二列起為測試資料，測試資料每一列為一用字串表示的方程式。

輸出說明：

對每一列的用字串表示的方程式，分別以一行輸出，求出  $x$  的解。 $x$  有三種情況：一個整數解、無窮多個解(IS)和無解(NS)。

輸入檔案 1：【檔名：in1.txt】

4

$x+5-3+x=7+x-2$

$2x=x$

$2x+3x-6x=x+4$

$2x=6$

輸入檔案 2：【檔名：in2.txt】

2

x=x

x=x+2

輸出範例：【檔名：out.txt】

3

0

-2

3

IS

NS

**子題 2：**將一個數字每個位數相加，直到剩個位數為止。(程式執行限制時間: 2 秒)

範例：

number= 39，則  $3+9 = 12$ ， $1+2 = 3$ ，3 是個為數，輸出 3。

number=65535，則  $6+5+5+3+5 = 24$ ， $2+4 = 6$ ，6 是個為數，輸出 6。

**輸入說明：**

第一列的數字  $n$  代表有幾筆資料要測試， $2 \leq n \leq 20$ 。第二列起為測試資料，測試資料每一列為一個數字(整數) $0 \leq N \leq 65535$ 。

**輸出說明：**

對每一列的數字，分別以一行輸出，將一個數字每個位數相加，直到剩個位數為止。

**輸入檔案 1：【檔名：in1.txt】**

3  
38  
39  
11111

**輸入檔案 2：【檔名：in2.txt】**

2  
8888  
65535

**輸出範例：【檔名：out.txt】**

2  
3  
5  
  
5  
6

## Problem 4：資料結構—樹

### 子題 1：樹。(程式執行限制時間: 2 秒)

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。

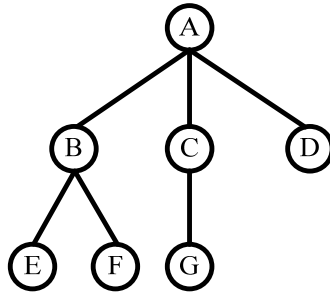


圖 4.1.1

在圖 4.1.1 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 4.1.1，因最大階度階度為 3，則其樹的高度為 3。

寫一個程式，讀入一無向圖的資料，一棵樹是一個無向圖且沒有迴圈。在測試檔中，節點的編號不一定是連續的號碼。如果檢測的圖形有迴圈，則輸出造成迴圈的邊；若該圖是樹，則輸出 T；沒有迴圈又不是樹，則輸出 F。

輸入說明：

第一列的數字  $n$  代表共有幾組資料要測試， $2 \leq n \leq 5$ 。

第二列起每一行代表一組測試資料。每組測試資料代表一圖形，內容為邊的資料。每個邊以 2 個整數  $i, j$  表示， $0 \leq i, j \leq 20$  and  $i \neq j$ ，其中  $i$  和  $j$  為節點的編號，代表從  $i$  節點和  $j$  節點有邊相連，每組測試資料，同一列中，每個邊的資料以空白( ) 隔開，而空白不限定一個， $|i, j|$  為邊的個數， $2 \leq |i, j| \leq 20$ 。

### 輸出說明：

每組測試資料輸出一列。測試資料的每個邊依序加入圖形，輸出造成迴圈的邊，測試資料拿掉這些造成迴圈的邊之後為一棵樹。如果檢測的圖形有迴圈，則輸出造成迴圈的邊；若該圖是樹，則輸出 T；沒有迴圈又不是樹，則輸出 F。

### 輸入檔案 1：【檔名：in1.txt】

4

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

8,1 1,3 6,2 8,10 7,5 1,4 7,8 8,6 8,0

3,8 6,8 6,4 0,6 8,2 2,0 5,3

1,0 4,3 1,2

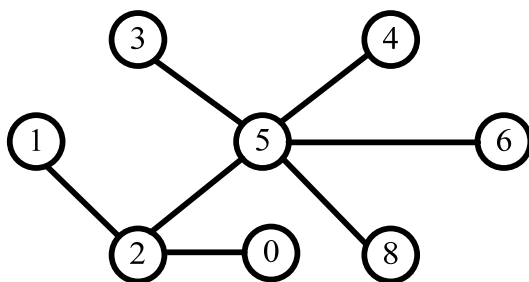


圖 4.1.1.1 (in1.txt)

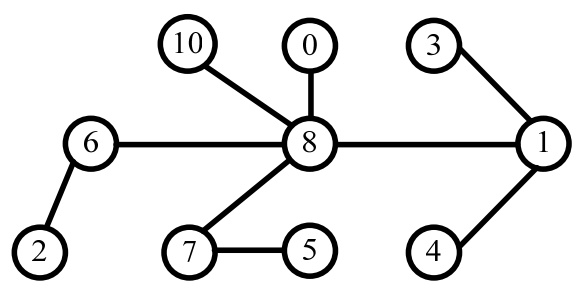


圖 4.1.1.2 (in1.txt)

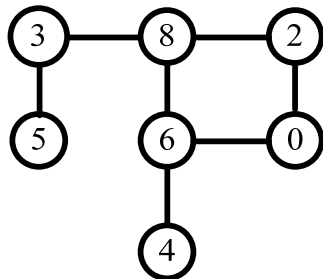


圖 4.1.1.3 (in1.txt)

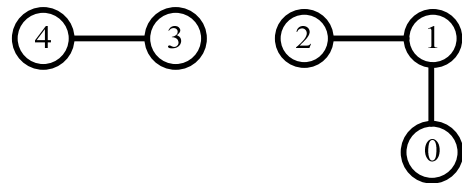


圖 4.1.1.4 (in1.txt)

輸入檔案 2：【檔名：in2.txt】

5  
1,2 2,3 4,0  
4,3 2,3 2,1 1,0  
1,2 2,3 3,4 1,4 1,5  
1,2 1,3 2,3  
1,2 2,3 3,4 1,4 1,5 5,4

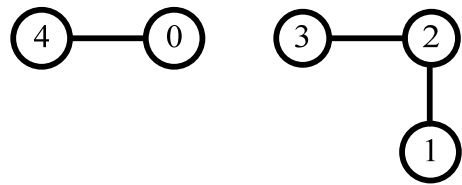


圖 4.1.2.1( in2.txt)

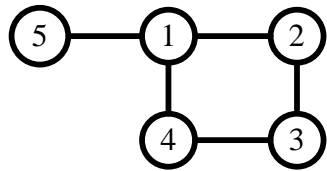


圖 4.1.2.3 ( in2.txt)

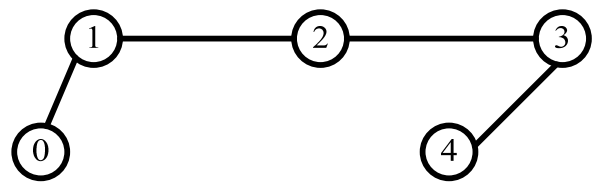


圖 4.1.2.2 ( in2.txt)

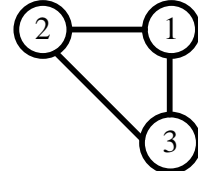


圖 4.1.2.4 (in2.txt)

輸出範例：【檔名：out.txt】

T  
T  
2,0  
F  
  
F  
T  
1,4  
2,3  
1,4 5,4

子題 2：樹葉節點到根節點之路徑，中間所經過的節點集合。(程式執行限制時間: 2 秒)

寫一個程式，讀入一樹狀結構的資料，依每組測試資料找出不包含根節點的所有**樹葉節點**，到根節點之路徑，各路徑長度為中間所經過的節點形成的集合之元素個數加 1。若中間所經過的節點集合為空集合則路徑長度為 1。請輸出每組測試資料中，每個**樹葉節點**到根節點之路徑**中間所經過的節點集合**，若**樹葉節點**到根節點之路徑長度為 1 時則輸出 N。

#### 輸入說明：

第一列的數字  $n$  代表共有幾組資料要測試， $2 \leq n \leq 5$ 。第二列起則是每一組測試資料。每組測試資料代表一個樹狀結構，每組測試資料中的以 0 為結束。每個邊以 2 個整數  $i, j$  表示， $0 \leq i, j \leq 80$  and  $i \neq j$ ， $i, j$  為節點的編號，代表  $i$  節點和  $j$  節點相連的一個邊，且節點  $j$  為節點  $i$  的父節點。若  $j$  為 99，則  $i$  為這組測試資料的根節點。輸入資料中，邊的資料依節點編號由小到大依序描述，即  $i$  的值會 0,1,2,...,  $m-1$  遞增， $m$  為節點的個數。

#### 輸出說明：

計算每組測試資料中，輸出每組測試資料中，每個樹葉節點到根節點之路徑中間所經過的節點集合，每組測試資料之後輸出換行。

在 in1.txt 檔案中，第一組測試資料中，樹葉節點為{1,2,4}；

則輸出樹葉節點{1,2,4}到根節點之路徑中間所經過的節點集合。依樹葉節點的數字大小由小大順序輸出，用集合的方式，紀錄樹葉節點到根節點之路徑中間所經過的節點，路徑紀錄方式，依這唯一的路徑，中間經過的節點之先後順序，樹葉節點 1 到根節點 0 之路徑為  $1 \rightarrow 3 \rightarrow 5 \rightarrow 0$ ，則中間所經過的節點集合為{3,5}，樹葉節點 1 和這集合，以 “:” 分隔。這組測試資料輸出如下：

1:{3,5}

2:{3,5}

4:{6,5}

在一行空行之後為下一組的測試資料的輸出結果。

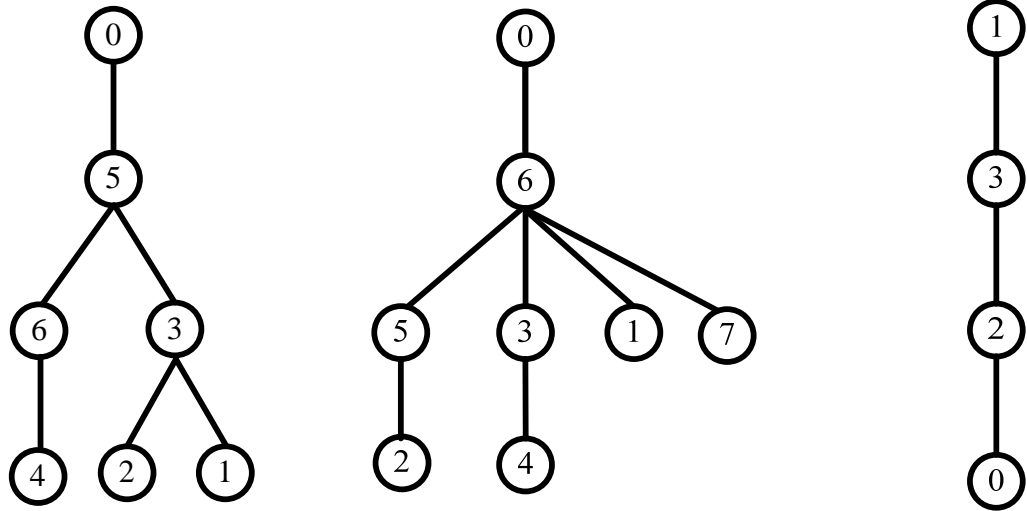
第二組測試資料中，樹葉節點為{1,2,4,7}；第三組測試資料中，樹葉節點的節點集合為{0}。輸出每個樹葉節點到根節點之路徑中間所經過的節點集合。

在 in2.txt 檔案的第一組測試資料中，樹葉節點為{1,2,3}；第二組測試資料中，樹葉節點集合為{1,2}。輸出每個樹葉節點到根節點之路徑中間所經過的節點集合。在一行空行之後為下一組的測試資料的輸出結果。

輸入檔案 1：【檔名：in1.txt】

3  
0,99  
1,3  
2,3  
3,5  
4,6  
5,0  
6,5  
0  
0,99  
1,6  
2,5  
3,6  
4,3  
5,6  
6,0  
7,6  
0  
0,2  
1,99  
2,3  
3,1  
0

三組輸入的資料所對應到的樹狀結構。由左而右，分別代表第 1、2 及 3 組資料。





輸入檔案 2：【檔名：in2.txt】

2  
0,99  
1,0  
2,0  
3,0  
**0**  
0,99  
1,0  
2,0  
**0**

輸出範例：【檔名：out.txt】

1:{3,5}  
2:{3,5}  
4:{6,5}

1:{6}  
2:{5,6}  
4:{3,6}  
7:{6}

0:{2,3}

1:N  
2:N  
3:N

1:N  
2:N

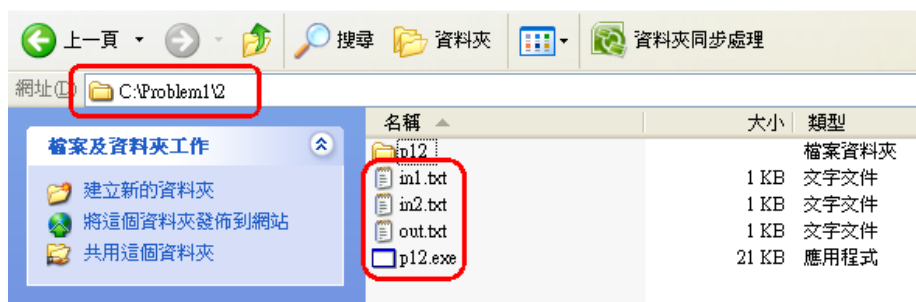
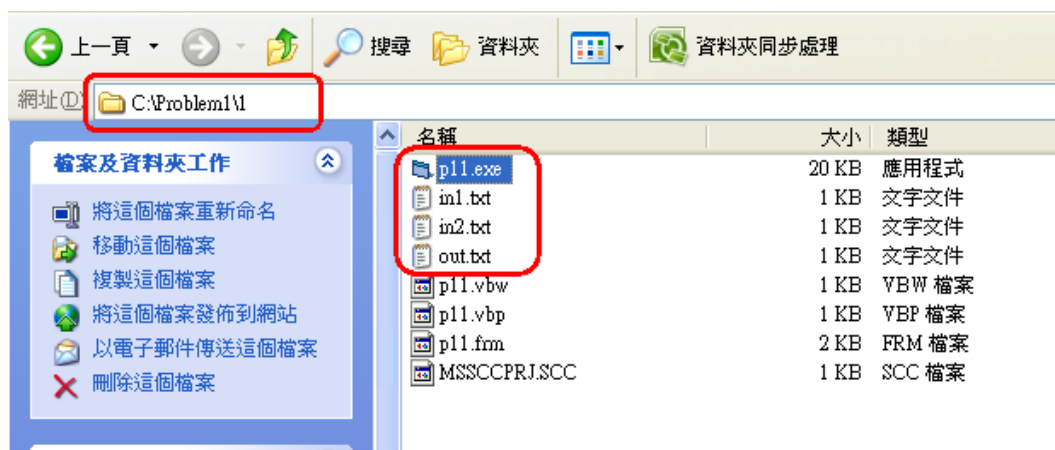
二組輸入的資料所對應到的樹狀結構。由左而右，分別代表第 1 及 2 組資料。



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
Problem1	子題 1	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
	子題 2	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
Problem2	子題 1	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
	子題 2	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
Problem3	子題 1	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
	子題 2	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
Problem4	子題 1	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
	子題 2	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