



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】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。

| 題目 | 子題 | 題目內容 | 配分 |
|----------|------|------------------|----|
| Problem1 | 子題 1 | 計算含有 s 或 S 字母的字數 | 11 |
| | 子題 2 | 給一個羅馬數字符號，轉為整數數字 | 13 |
| Problem2 | 子題 1 | 信用卡卡號 | 10 |
| | 子題 2 | 幾 A 幾 B | 12 |
| Problem3 | 子題 1 | 網段廣播位址 | 9 |
| | 子題 2 | 大數排序問題 | 14 |
| Problem4 | 子題 1 | 樹 | 16 |
| | 子題 2 | 後序運算式 | 15 |

光碟有試題電子檔，選手可參考。

線上評分系統 vbjudge.fedora.idv.tw

Problem 1：字串問題

子題 1：計算含有 s 或 S 字母的字數。（程式執行限制時間: 2 秒) **11 分**

每列測試資料(每組測試資料)有多個英文字，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。寫一程式計算每列測試資料(每組測試資料)含有 s 或 S 字母的字數。

輸入說明：

第 1 列的數字 n 代表有幾組資料要測試， $2 \leq n \leq 20$ ，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 ≤ 255 。

輸出說明：

每組測試資料輸出一列，計算每列含有 s 或 S 字母的字數。

輸入檔案 1：【檔名：in1.txt】

3

This is a sample file.

Hello World!!

SOS!

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 spreaDitng unchecked over its face.

輸出範例：【檔名：out.txt】

3

0

1

2

5

9

子題 2：給一個羅馬數字符號，轉為整數數字。(程式執行限制時間: 2 秒) **13 分**

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。數碼有限制，同樣數碼最多只能出現 3 次，例如 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羅馬數字符號，這個羅馬數字符號所對應到的整數數字會落在 1 到 3999 之間。

輸出說明：

將每筆的測試資料轉為整數數字。

輸入檔案 1 :【檔名：in1.txt】

7

I

V

X

L

C

D

M

輸入檔案 2 :【檔名：in2.txt】

8

IX

VI

XI

XVII

CCLXVIII

MMMCC

DCCVII

MMCDLXIX

輸出範例：【檔名：out.txt】

1

5

10

50

100

500

1000

9

6

11

17

268

3200

707

2469

Problem 2：信用卡卡號、幾 A 幾 B

子題 1：信用卡卡號。(程式執行限制時間: 2 秒) **10 分**

信用卡卡號是經由 Luhn 算法所產生出來的，以 1234 5678 9123 4563 為範例，由右至左（共 1 ~ 16）；碼數奇數位乘積為 1(x1)，碼數偶數位乘積為 2(x2)，共得出 16 個[新數]；將每個[新數]的十位數加上個位數，再產生[新新數]，共 16 個[新新數] (另外的一個說法：如有位數的值變為雙位數 ≥ 10 ，則把該位數的值減 9)；把所有 16 個[新新數]合計加總，能整除 10 者，則該組為有效的信用卡號碼。

以 1234 5678 9123 4563 為範例

| 字串 index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------------|----|----|----|----|-----------|----|-----------|----|-----------|----|----|----|----|----|-----------|----|
| 碼數 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 卡號 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 | 4 | 5 | 6 | 3 |
| 乘積 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 |
| 新數 | 2 | 2 | 6 | 4 | <u>10</u> | 6 | <u>14</u> | 8 | <u>18</u> | 1 | 4 | 3 | 8 | 5 | <u>12</u> | 3 |
| 新新數 | 2 | 2 | 6 | 4 | <u>1</u> | 6 | <u>5</u> | 8 | <u>9</u> | 1 | 4 | 3 | 8 | 5 | <u>3</u> | 3 |

把所有的新新數相加 $2 + 2 + 6 + 4 + 1 + 6 + 5 + 8 + 9 + 1 + 4 + 3 + 8 + 5 + 3 + 3 = 70$

以 4013 7356 3380 0642 為範例

| 字串 index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------------|----|----|----|----|-----------|----|-----------|----|----|----|-----------|----|----|----|----|----|
| 碼數 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 卡號 | 4 | 0 | 1 | 3 | 7 | 3 | 5 | 6 | 3 | 3 | 8 | 0 | 0 | 6 | 4 | 2 |
| 乘積 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 |
| 新數 | 8 | 0 | 2 | 3 | <u>14</u> | 3 | <u>10</u> | 6 | 6 | 3 | <u>16</u> | 0 | 0 | 6 | 8 | 2 |
| 新新數 | 8 | 0 | 2 | 3 | <u>5</u> | 3 | <u>1</u> | 6 | 6 | 3 | <u>7</u> | 0 | 0 | 6 | 8 | 2 |

把所有的新新數相加 $8 + 0 + 2 + 3 + 5 + 3 + 1 + 6 + 6 + 3 + 7 + 0 + 0 + 6 + 8 + 2 = 60$

以 5181 2710 9900 0017 為範例

| 字串 index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------------|-----------|----|-----------|----|----|----|----|----|-----------|----|----|----|----|----|----|----|
| 碼數 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 卡號 | 5 | 1 | 8 | 1 | 2 | 7 | 1 | 0 | 9 | 9 | 0 | 0 | 0 | 0 | 1 | 7 |
| 乘積 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 | x2 | x1 |
| 新數 | <u>10</u> | 1 | <u>16</u> | 1 | 4 | 7 | 2 | 0 | <u>18</u> | 9 | 0 | 0 | 0 | 0 | 2 | 7 |
| 新新數 | <u>1</u> | 1 | <u>7</u> | 1 | 4 | 7 | 2 | 0 | <u>9</u> | 9 | 0 | 0 | 0 | 0 | 2 | 7 |

把所有的新新數相加 $1 + 1 + 7 + 1 + 4 + 7 + 2 + 0 + 9 + 9 + 0 + 0 + 0 + 0 + 2 + 7 = 50$

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，測試資料為信用卡 16 位卡號。

輸出說明：

每列輸出 T 或 F，對應到每筆測試資料中為有效的信用卡卡號 T 或無效的信用卡卡號 F。

輸入檔案 1：【檔名：in1.txt】

```
5
1234567891234563
4013735633800642
5181271099000017
5241150318192904
5241150313182900
```

輸入檔案 2：【檔名：in2.txt】

```
3
1234567891234561
4013735633800643
5181271099000014
```

輸出範例：【檔名：out.txt】

```
T
T
T
T
T

F
F
F
```

子題 2：幾 A 幾 B。(程式執行限制時間: 2 秒) 12 分

遊戲規則設定：數字個數為 2~6，為不同數字。

幾 A 幾 B 的判斷方式：『? A ? B』，A 代表數值及位置都相同，B 表示只有數值相同但位置不同。

例如 5234/5789，結果就是 1A0B，5 的數值及位置都對了，234 和 789 這三個數值都不相同。

例如 5634/6589，結果就是 0A2B，5 和 6 的數值都對，但位置都不同。

例如 1847/6149，結果就是 1A1B。

例如 0234/0742，結果就是 1A2B。

例如 54321/01234，結果就是 0A4B。

例如 123678/023687，結果就是 3A2B。

在排列組合問題中將一組數字進行排列，可以得到不同的數字順序，例如 12 這個數的排列共有：(1)12、(2)21 二組(由小到大排序)；例如 123 這個數的排列組合順序為：(1)123、(2)132、(3)213、(4)231、(5)312、(6)321 六組(由小到大排序)；例如 1234 這數的排列組合有 24 組，數列順序如下：

| | | | |
|---------|----------|----------|----------|
| (1)1234 | (7)2134 | (13)3124 | (19)4123 |
| (2)1243 | (8)2143 | (14)3142 | (20)4132 |
| (3)1324 | (9)2314 | (15)3214 | (21)4213 |
| (4)1342 | (10)2341 | (16)3241 | (22)4231 |
| (5)1423 | (11)2413 | (17)3412 | (23)4312 |
| (6)1432 | (12)2431 | (18)3421 | (24)4321 |

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，共有三個正整數 i, j, k 。各個數字間以“,” 隔開。 i 的值為 12, 123, 1234, 12345, 123456 其中之一，而 j 和 k 代表 i 值排列組合順序(由小到大排序)的第 j 個和第 k 個值。 j 和 k 不會超出 i 的排列數，請輸出第 j 個和第 k 個值的幾 A 幾 B。

例如 i, j, k : 12, 1, 2 這組測試資料中，12 這個數的排列組合有：(1)12、(2)21 二組(由小到大排序)，第 1 個值為 12；第 2 個值為 21；幾 A 幾 B 為 0A2B。

例如 i, j, k : 123, 1, 2 這組測試資料中，123 這個數的排列組合有：(1)123、(2)132、(3)213、(4)231、(5)312、(6)321 六組(由小到大排序)，第 1 個值為 123；第 2 個值為 132；幾 A 幾 B 為 1A2B。

例如 i, j, k : 123, 2, 5 這組測試資料中，123 這個數的排列組合有：(1)123、(2)132、(3)213、(4)231、(5)312、(6)321 六組(由小到大排序)，第 2 個值為 132；第 5 個值為 312；幾 A 幾 B 為 1A2B。

例如 i, j, k : 1234, 15, 9 這組測試資料中，1234 這個數的排列組合有 24 組(由小到大排序)，第 15 個值為 3214；第 9 個值為 2314；幾 A 幾 B 為 2A2B。

例如 i, j, k : 1234, 3, 4 這組測試資料中，1234 這個數的排列組合有 24 組(由小到大排序)，第 3 個值為 1324；第 4 個值為 1342；幾 A 幾 B 為 2A2B。

輸出說明：

每組測試資料輸出一列。輸出以 i 值排列組合順序中，找出第 j 個和第 k 個的值，再算出個這二個值的幾 A 幾 B。對於每組資料，輸出『 ? A ? B』，A 代表數值及位置都相同，B 表示數值相同但位置不同。

輸入檔案 1：【檔名：in1.txt】

3
12,1,2
123,1,2
123,2,5

輸入檔案 2：【檔名：in2.txt】

4
1234,15,9
1234,3,4
1234,1,24
1234,1,1

輸出範例：【檔名：out.txt】

0A2B
1A2B
1A2B

2A2B
2A2B
0A4B
4A0B

Problem 3：數學問題

子題 1：網段廣播位址 (程式執行限制時間: 2 秒) 9 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廣播位址呢？

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，來求出網段廣播位址。我們任選一個 IP 位址來做例子：

範例一：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(NOT 子網路遮罩)以 OR 運算：

10001011.10101111.10011001.11111100

OR

00000000.00000000.11111111.11111111

得出網段廣播位址：

10001011.10101111. 11111111.11111111

10001011.10101111.11111111.11111111 換成十進位就是 139.175.255.255，就是網段廣播位址。

範例二：網段廣播位址:= IP 位址 OR (NOT 子網路遮罩) = 10.104.69.0 OR (NOT 255.255.255.192) = 10.104.69.0 OR 0.0.0.63 = 10.104.69.63

範例三：假設某 IP 位址為 192.15.156.205，子網路遮罩為 255.255.255.224，請推算出該網段廣播位址？

先將子網路遮罩 255.255.255.224 做一個 NOT 運算：

11111111.11111111.11111111.11100000 => 00000000.00000000.00000000.00011111

然後再和 IP 做一次 OR 運算，就可以得到網段廣播位址：

192.15.156.205 OR (NOT 255.255.255.224)

11000000.00001111.10011100.11001101

OR 00000000.00000000.00000000.00011111

得出：11000000.00001111.10011100.11011111

換成十進位就：192.15.156.223

範例四和五：測試資料會有 IP 位址/子網路遮罩資訊，寫一程式計算網段廣播位址。

| | | |
|-----------|---------------------------|---|
| 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 |
| NOT 子網路遮罩 | <u>0.0.0.31</u> (修正說明) | <u>00000000. 00000000. 00000000.000 11111</u> (修正說明) |
| 網段廣播位址 | 192.168.10.95 | 11000000.10101000.00001010.010 11111 |

| | | |
|-----------|-----------------------------|--|
| 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 |
| NOT 子網路遮罩 | <u>0.0.63.255</u> (修正說明) | <u>00000000. 00000000.00 111111.11111111</u> (修正說明) |
| 網段廣播位址 | 10.240.191.255 | 00001010.11110000.10 111111.11111111 |

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，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，輸出網段廣播位址。

輸入檔案 1：【檔名：in1.txt】

5

139.175.153.252/255.255.0.0

10.104.69.0/255.255.255.192

192.15.156.205/255.255.255.224

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】

139.175.255.255

10.104.69.63

192.15.156.223

192.168.10.95

10.240.191.255

172.16.115.255

192.168.168.31

輸入檔案 1：【檔名：in1.txt】

4

[illegible]

輸入檔案 2：【檔名：in2.txt】

3

4, 2, 1, 5

1, 2, 3, 4, 5, 6

7, 6, 5, 4, 3, 2

輸出範例：【檔名：out.txt】

2, 3, 1, 5, 4

3, 1, 2

2, 1, 3

4, 1, 2, 3

3, 2, 1, 4

1, 2, 3, 4, 5, 6

6, 5, 4, 3, 2, 1

Problem 4：資料結構—樹、後序法

子題 1：樹。(程式執行限制時間: 2 秒) 16 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。

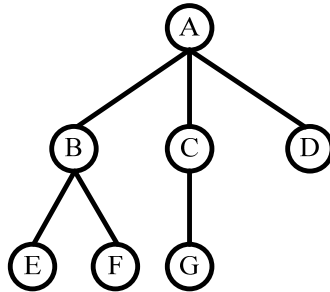


圖 4.1.1

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。在測試檔中，節點的編號不一定是連續的號碼。

輸入說明：

第一列的數字 n 代表共有幾組資料要測試， $2 \leq n \leq 20$ 。

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，每個邊的資料以一個或多個空白隔開， $|i, j|$ 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。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。如果檢測的圖形有迴圈(最多一個迴圈)，則輸出造成迴圈的所有點(由小到大依序輸出)；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

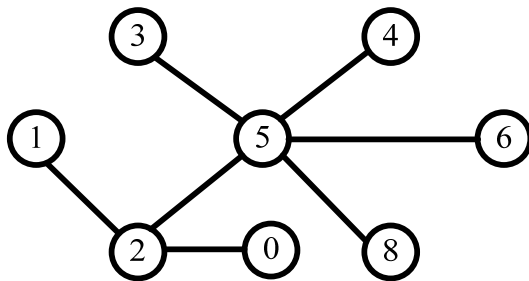


圖 4.1.1.1 (in1.txt)

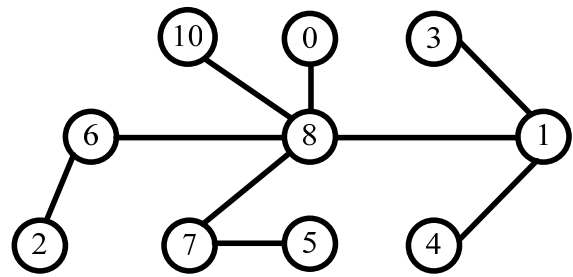


圖 4.1.1.2 (in1.txt)

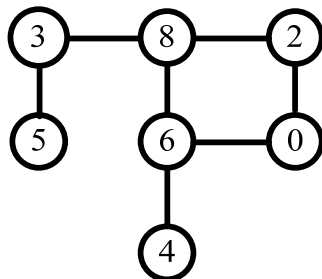


圖 4.1.1.3 (in1.txt)

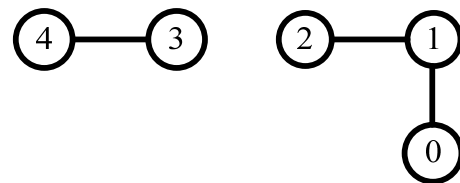


圖 4.1.1.4 (in1.txt)

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 1,5 5,4

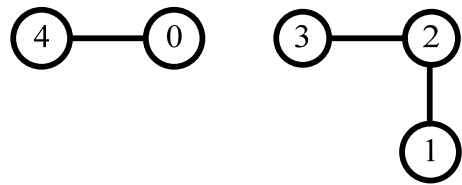


圖 4.1.2.1 (in2.txt)

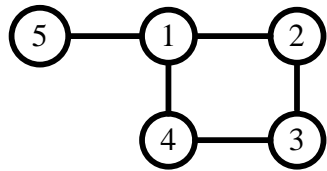


圖 4.1.2.3 (in2.txt)

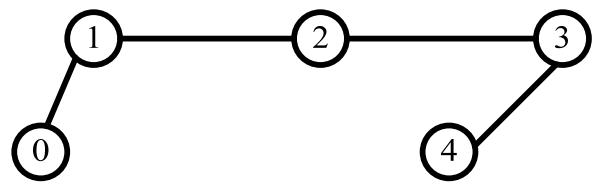


圖 4.1.2.2 (in2.txt)

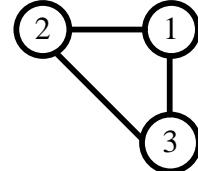


圖 4.1.2.4 (in2.txt)

輸出範例：【檔名：out.txt】

T
T
0, 2, 6, 8
F

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

子題 2：後序運算式(postfix)。(程式執行限制時間: 2 秒) 15 分

求一個運算式值，一般在電腦中，常用的運算式表示法有下列三種表示法：

(1)中序運算式(infix) :<運算元> <運算子> <運算元>，如 $A+B$ 。

(2)前序運算式(prefix) :<運算子> <運算元> <運算元>，如 $+AB$ 。

(3)後序運算式(postfix) :<運算元> <運算元> <運算子>，如 $AB+$ 。

後序運算式(postfix)有別於我們習慣的運算式寫法(中序運算式,infix)，是把運算子寫在運算元之後，雖然對人來說可讀性很低，可是對電腦是很方便的。後序運算式的好處還可以不用考慮中序運算式的先乘除後加減的問題。

比如我們習慣 $3 + 5$ 這樣的式子，就是中序運算式，同一個式子改成後序寫法即為 $3 5 +$

輸入後序運算式，求出運算式值。

後序表示法 $9 2 3 5 + * +$ 對應到中序運算式 $9 + 2 * (3 + 5)$

輸出: 25

提示： 後序運算式的演算法

(1)從後序運算式字串中取出一<運算元>或<運算子>，分析之。

(2)若為運算元，則將該運算元放入(PUSH)運算元堆疊。

若為運算子，則由運算元堆疊中連續彈出(POP)兩個運算元，先彈出者(POP)為 B ，後彈出者(POP)為 A ，再依 “ A 運算子 B ”之運算式求得其值，運算式結果放入(PUSH)運算元堆疊。

(3)回到(1)，直到後序運算式字串完全處理完畢。

彈出(POP)運算元堆疊內之值，即為答案。

輸入說明：

第一列的數字 n 代表有幾筆資料要測試， $2 \leq n \leq 20$ ，第二列起為每組的測試資料，之後每一列為每筆的測試資料。輸入一個後序運算式的字串包含數個運算元及運算子，為了方便讀取，所有的運算元與運算子之間以一個或多個空白隔開。運算子包含 $+$ $-$ $*$ $/$ 等四個，運算元則為 $0 \sim 2^{31} - 1$ 的整數。

輸出說明：

每組測試資料輸出該後序運算式的結果。

為避免小數誤差問題，運算結果必定為整數，運算過程中不會出現小數的結果，不會有 $4/3$ 或 $3/2$ 無法整除的測試資料，因此請放心使用整數來進行運算。也不會有溢位運算。

輸入檔案 1：【檔名：in1.txt】

```
3
3 5 +
9 2 3 5 + * +
6 3 / 1 4 - * 3 + 8 -
```

輸入檔案 2：【檔名：in2.txt】

```
3
2 3 + 4 *
2 3 4 + * 5 *
2 123 -
```

輸出範例：【檔名：out.txt】

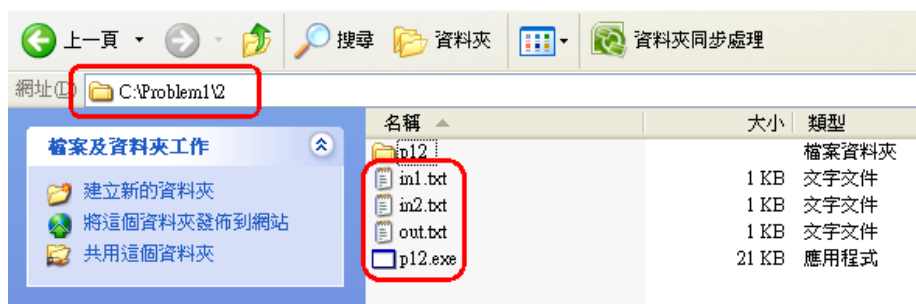
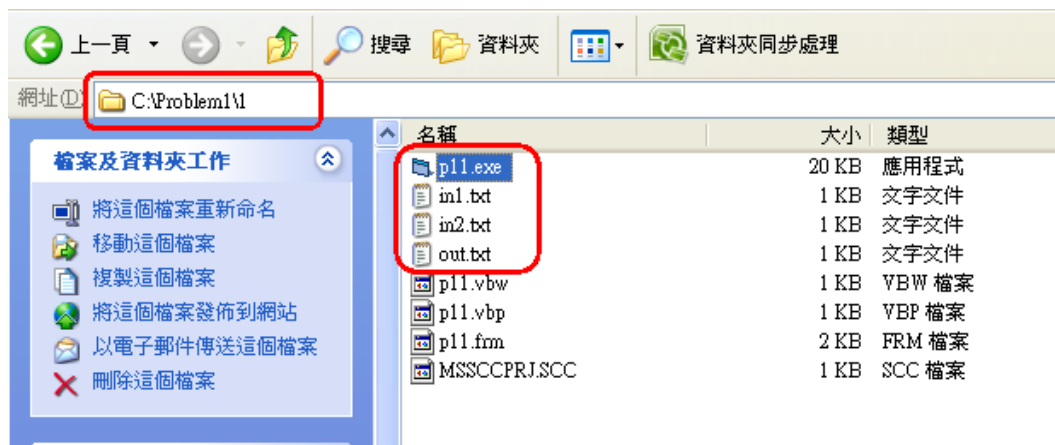
```
8
25
-11

20
70
-121
```

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 |