



# 全國高級中等學校 104 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	電梯電費計算系統	9
	子題 2	樂透	12
Problem2	子題 1	排列組合	13
	子題 2	最大公約數計算	11
Problem3	子題 1	計算位元為 1 的個數	10
	子題 2	矩陣的乘法 $AB = A \times B$	15
Problem4	子題 1	輸出二元樹的後序拜訪的結果	14
	子題 2	最小成本生成樹	16

### Problem 1：生活問題

子題 1：電梯電費計算系統。(程式執行限制時間: 2 秒) 9 分

假設你身為一個電梯公司的工程師，正要為某個百貨公司的電梯設計一套電費計算系統，計算百貨公司的電梯某時段所耗的電費是多少。以下是電梯所耗電力之電費的規則：

- (1) 電梯上樓時，每經過一個樓層，所需電費為 20 元。
- (2) 電梯下樓時，每經過一個樓層，所需電費為 10 元。
- (3) 假設電梯停在某一個樓層時不會耗電。

舉例來說：電梯從 3 樓到 8 樓再到 5 樓，則所耗的電費為：從 3 樓到 8 樓，所耗的電費是  $(8 - 3) \times 20 = 100$  元；電梯從 8 樓到 5 樓，所耗的電費是  $(8 - 5) \times 10 = 30$  元。所以總共花了 130 元。

#### 輸入說明：

第一列的數字  $n$  代表有幾組資料要測試， $1 \leq n \leq 5$ ，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。每組測試資料第一列是一個整數  $2 \leq x \leq 10$ ，用來表示某時段電梯所停過的樓層數；每組測試資料第二列是一組以“,”分隔的  $x$  個數字(相鄰的數字不會相同)，分別表示電梯先後停過的樓層。輸入測試資料的電梯樓層最高不會超過 20 樓(含)。

#### 輸出說明：

每組測試資料輸出一列。請根據電梯上下運作的樓層，計算出某時段電梯運作所花的電費。

#### 輸入檔案 1：【檔名：in1.txt】

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

#### 輸入檔案 2：【檔名：in2.txt】

```
2
3
18,19,20
3
2,5,2
```

#### 輸出範例：【檔名：out.txt】

```
130
310

40
90
```

## 子題 2：樂透。(程式執行限制時間: 2 秒) 12 分

今彩 539 是一種樂透型遊戲，投注者必須從 01~39 的號碼中任選 5 個不同的號碼進行投注。開獎時，開獎單位將隨機開出五個號碼，這一組號碼就是該期今彩 539 的中獎號碼，也稱為「獎號」。投注者的五個選號中，如有二個以上(含二個號碼)對中當期開出之五個中獎號碼，即為中獎。

為了方便包牌，投注者可任選 6 個不同號碼，共可以產生 6 組不同的 5 個號碼組合。用程式計算投注者任選的 6 個不同號碼的包牌的組合，各對中 2 碼(對中當期獎號之其中任二碼)、3 碼、4 碼及 5 碼的次數。

### 輸入說明：

第一列的數字  $n$  代表有幾組資料要測試， $1 \leq n \leq 5$ 。第二列為該期今彩 539 的中獎號碼(獎號)，已由小到大排序好。第三列起為測試資料，每組測試資料為一行，有 6 個數字，是投注者任選的 6 個不同號碼，各個號碼間以“,”隔開，已由小到大排序好。

### 輸出說明：

計算每組測試資料中，獎號和投注者的 6 個號碼，用程式計算投注者任選的 6 個不同號碼所組合的 6 種不同投注，對中 2 碼(對中當期獎號之其中任二碼)、3 碼、4 碼及 5 碼各中幾次，各個對中次數間以“,”隔開，且輸出順序依規定。

例如今彩 539 的中獎號碼為 01,07,28,29,30：投注者任選的 6 個不同號碼為 01,07,29,30,36,39  
6 種不同投注為：(底線是為了方便比對中獎號碼)

07,29,30,36,39 對中 3 碼

01,29,30,36,39 對中 3 碼

01,07,30,36,39 對中 3 碼

01,07,29,36,39 對中 3 碼

01,07,29,30,39 對中 4 碼

01,07,29,30,36 對中 4 碼

在這 6 個不同號碼對中 2 碼 0 次；3 碼 4 次；4 碼 2 次；5 碼 0 次；則輸出 0,4,2,0。

例如今彩 539 的中獎號碼為 01,07,28,29,30：投注者任選的 6 個不同號碼為 01,07,22,23,24,38  
6 種不同投注為：

07,22,23,24,38 對中 1 碼

01,22,23,24,38 對中 1 碼

01,07,23,24,38 對中 2 碼

01,07,22,24,38 對中 2 碼

01,07,22,23,38 對中 2 碼

01,07,22,23,24 對中 2 碼

在這 6 個不同號碼對中 2 碼 4 次；3 碼 0 次；4 碼 0 次；5 碼 0 次；則輸出 4,0,0,0。

例如今彩 539 的中獎號碼為 21,22,23,24,32：投注者任選的 6 個不同號碼為 21,22,23,24,25,32  
6 種不同投注為：

22,23,24,25,32 對中 4 碼

21,23,24,25,32 對中 4 碼

21,22,24,25,32 對中 4 碼

21,22,23, 25,32 對中 4 碼

21,22,23,24, 32 對中 5 碼

21,22,23,24,25 對中 4 碼

在這 6 個不同號碼對中 2 碼 0 次；3 碼 0 次；4 碼 5 次；5 碼 1 次；則輸出 0,0,5,1

。

輸入檔案 1：【檔名：in1.txt】

2

01,07,28,29,30

01,07,29,30,36,39

01,07,22,23,24,38

輸入檔案 2：【檔名：in2.txt】

2

21,22,23,24,32

01,02,03,04,05,06

21,22,23,24,25,32

輸出範例：【檔名：out.txt】

0,4,2,0

4,0,0,0

0,0,0,0

0,0,5,1

## Problem 2：數學問題

子題 1：排列組合。(程式執行限制時間: 3 秒) 13 分

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， $2 \leq n \leq 5$ ，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  $k$  個值的總合。

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；總合為  $12+21=33$ 。

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；總合為  $123+132=255$ 。

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；總合為  $132+312=444$ 。

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；總合為  $3214+2314=5528$ 。

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；總合為  $1324+1342=2666$ 。

### 輸出說明：

每筆測試資料輸出一列。輸出以  $i$  值排列組合順序中，找出第  $j$  個和第  $k$  個的值，再算出這二個值的總合。

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】

2

1234,15,9

1234,3,4

輸出範例 :【檔名：out.txt】

33

255

444

5528

2666

**子題 2：**最大公約數計算。(程式執行限制時間: 2 秒) **11 分**

最大公因數 (Greatest Common Divisor, 簡寫為 G.C.D.)，指某幾個正整數共有因數中最大的一個。例如數字 20,8 和 30 的最大公因數為 2。

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

$$8 = 2^3$$

$$30 = 2^1 \times 3^1 \times 5^1$$

$$\text{GCD}(20,8,30)=2$$

**輸入說明：**

第一列的數字  $n$  代表有幾筆資料要測試， $1 \leq n \leq 5$ ，第二列起為測試資料，之後每列為每筆測試資料，每筆測試資料至少有 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。

**輸出說明：**

每筆測試資料輸出一列。算出每列這幾個正整數的最大公因數。

**輸入檔案 1：【檔名：in1.txt】**

3

20,8,30

10,20,30

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】**

2

10

4

12

1

1

### Problem 3：其他

子題 1：計算位元為 1 的個數。(程式執行限制時間: 2 秒) 10 分

計算機概論中的數字系統轉換，內容是將一個十進位的數字，轉換成二進位的數字。現在請你設計一個程式，計算由十進位數字(整數)轉換的二進位數字中，位元值為 1 的位元個數。

#### 輸入說明：

第一列的數字  $n$  代表有幾筆資料要測試， $1 \leq n \leq 5$ 。第二列起為測試資料，測試資料每一列為一個十進位數字(整數) $0 \leq N \leq 65535$ 。

#### 輸出說明：

對每一列的十進位數字，分別以一行輸出，計算轉換成二進位數字中，位元值為 1 的位元個數。

#### 輸入檔案 1：【檔名：in1.txt】

```
2
1025
65535
```

#### 輸入檔案 2：【檔名：in2.txt】

```
2
0
3
```

#### 輸出範例：【檔名：out.txt】

```
2
16

0
2
```



**子題 2：**矩陣的乘法 $AB = A \times B$ 。(程式執行限制時間: 2 秒) **15 分**

矩陣的乘法

若  $A$  是  $m \times r$  矩陣， $B$  是  $r \times n$  矩陣，那麼  $A$ 、 $B$  的乘積就是一個  $m \times n$  的矩陣，其乘積的第  $i$  列第  $j$  行的元素就是  $A$  中  $i$  列的元素與  $B$  中  $j$  行相對應的元素的乘積之和，我們以  $AB$  表示  $A$  和  $B$  的矩陣的乘積，則  $AB$  中  $i$  列  $j$  行的元素記作  $AB_{ij}$ ，其值為：

$$AB_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + a_{i3}b_{3j} + \dots + a_{ir}b_{rj}$$

請注意，只有當矩陣  $A$  的行數與  $B$  的列數相等的情況下才能進行矩陣的乘法運算。

例如

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -4 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 1 & 0 & 2 \\ -1 & 2 & 5 & 0 \\ 0 & -2 & 2 & 1 \end{bmatrix}$$

$$AB = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -4 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 0 & 2 \\ -1 & 2 & 5 & 0 \\ 0 & -2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 16 & 5 \\ 4 & -10 & -18 & 1 \end{bmatrix}$$

- $AB$  第一列第一行的元素就是  $A$  的第一列的元素與  $B$  的第一行的元素的乘積之和。那麼，  
 $AB_{11} = (1)(3) + (2)(-1) + (3)(0) = 1$
- $AB$  第一列第二行的元素就是  $A$  的第一列的元素與  $B$  的第二行的元素的乘積之和。那麼，  
 $AB_{12} = (1)(1) + (2)(2) + (3)(-2) = -1$
- $AB$  第二列第一行的元素就是  $A$  的第二列的元素與  $B$  的第一行的元素的乘積之和。那麼，  
 $AB_{21} = (0)(3) + (-4)(-1) + (1)(0) = 4$

$A$  是  $m \times r$  的矩陣， $B$  是  $r \times n$  的矩陣；則  $AB$  是  $m \times n$  的矩陣

A=	1	2	3
1	2	4	1
2	3	6	2
3	2	5	0
4	1	2	3

B=	1	2	3	4	5
1	2	6	2	0	2
2	3	1	1	1	2
3	4	2	2	0	1

AB=	1	2	3	4	5
1	20	18	10	4	13
2	32	28	16	6	20
3	19	17	9	5	14
4	20	14	10	2	9

$$AB_{11}=2*2+4*3+1*4=20$$

$$AB_{21}=3*2+6*3+2*4=32$$

$$AB_{31}=2*2+5*3+0*4=19$$

$$AB_{41}=1*2+2*3+3*4=20$$

#### 輸入說明：

第一列的數字  $x$  代表有幾筆資料要測試， $1 \leq x \leq 5$ ，第二列起為測試資料，每組測試資料為三個矩陣  $A$ 、 $B$  和  $AB$ ，每組測試資料第一列為 4 個數字  $m, r, r, n$  ( $2 \leq m, r, r, n \leq 8$ )，之後  $m + r + m$  列為三個矩陣  $A$ 、 $B$  和  $AB$  的內容，資料以空白( )隔開，而空白不限定一個，矩陣  $A$  和  $B$  的測試資料值為  $-20 \leq Z \leq 20$  的整數。矩陣  $A$  或  $B$  的測試資料中會有一個值為 9999 代表需要修正的值。

#### 輸出說明：

每組測試資料輸出一列，依據三個矩陣  $A$ 、 $B$  和  $AB$  的內容，修正矩陣  $A$  或  $B$  的測試資料中，值為 9999，找出 9999 在矩陣  $A$  或  $B$  原來的值。

輸入檔案 1 :【檔名：in1.txt】

2  
2,3,3,4  
1 2 9999  
0 -4 1  
3 1 0 2  
-1 2 5 0  
0 -2 2 1  
1 -1 16 5  
4 -10 -18 1  
4,3,3,5  
2 4 1  
3 6 2  
2 5 0  
1 2 3  
2 6 2 0 2  
3 1 1 1 2  
9999 2 2 0 1  
20 18 10 4 13  
32 28 16 6 20  
19 17 9 5 14  
20 14 10 2 9

輸入檔案 2 :【檔名：in2.txt】

2  
2,3,3,2  
1 1 1  
1 9999 1  
1 1  
1 1  
1 1  
3 3  
3 3  
2,2,2,2  
1 1  
1 0  
9999 -1  
-1 -1  
-2 -2  
-1 -1

輸出範例：【檔名：out.txt】

3

4

1

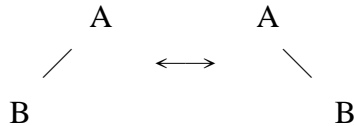
-1

#### Problem 4：資料結構—樹

子題 1：輸出二元樹的後序拜訪的結果。(程式執行限制時間: 2 秒) 14 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 preorder、中序 inorder 及後序 postorder。

後序 postorder 定義：

拜訪根節點前，若有左子樹，先拜訪其左子樹的所有節點；若有右子樹，再拜訪其右子樹的所有節點，最後再拜訪根節點。

二元搜尋樹(Binary Search Tree)定義：

二元搜尋樹是一種二元樹，它可以為空集合，若不為空集合，則必須要滿足以下條件：

- 1.若左子樹不為空集合，則左子樹的鍵值均須要小於樹根的鍵值。
- 2.若右子樹不為空集合，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。

由使用者輸入  $x$  筆資料，建立一個二元搜尋樹(Binary Search Tree)，輸出二元搜尋樹的後序拜訪的結果。

輸入說明：

第一列的數字  $n$  代表有幾組資料要測試， $1 \leq n \leq 5$ ，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是一個整數  $3 \leq x \leq 20$ ，用來表示這組測試資料有幾個節點；每組測試資料的第二列為這組測試資料各節點編號，以“,”分隔各節點編號，編號為一整數  $0 \leq N \leq 99$ ，各節點編號不會相同。用測試資料以二元搜尋樹方式建樹。

輸出說明：

在測試資料中所建二元搜尋樹，輸出二元搜尋樹的後序拜訪的結果，以“,”分隔各節點編號。

輸入檔案 1 :【檔名：in1.txt】

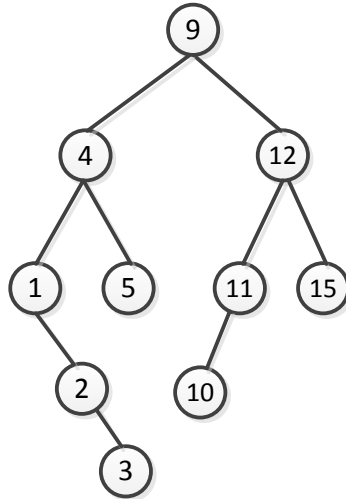
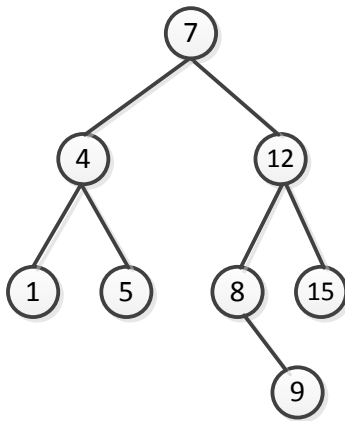
2

8

7,4,1,5,12,8,9,15

10

9,4,1,5,12,11,10,15,2,3



輸入檔案 2 :【檔名：in2.txt】

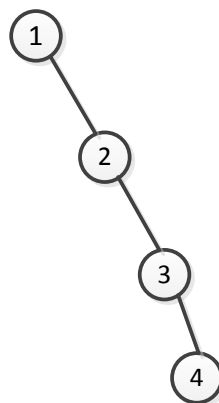
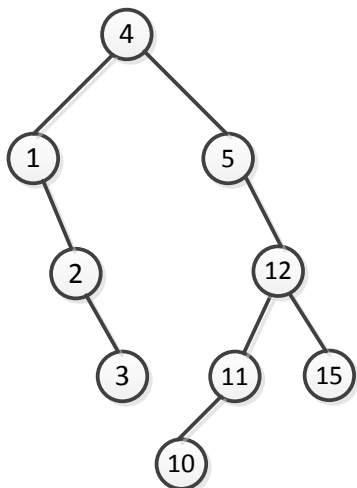
2

9

4,1,5,12,11,10,15,2,3

4

1,2,3,4



輸出範例 :【檔名：out.txt】

1,5,4,9,8,15,12,7

3,2,1,5,4,10,11,15,12,9

3,2,1,10,11,15,12,5,4

4,3,2,1

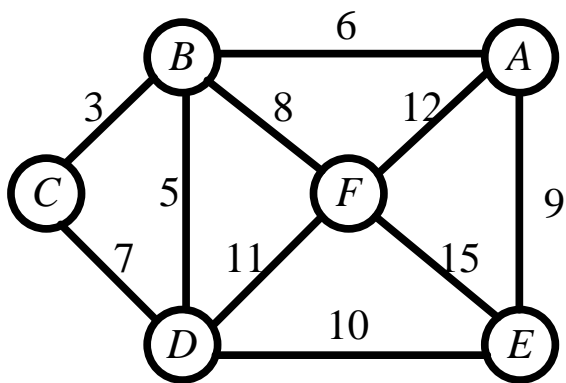
**子題 2：最小成本生成樹。(程式執行限制時間: 2 秒) 16 分**

以有線電視電纜的架設為例，若只能沿著街道佈線，則以街道為邊，而路口為節點，其中必然有一最小成本生成樹能使佈線成本最低。

給定一個圖形中，有許多條邊(線)連結了所有的節點，這些邊都有一個數值，代表此邊的成本。我們可以去掉圖形中的某些邊，使得剩下的邊能連結所有的節點，且邊的數量比節點的數量少 1，這些節點和留下的邊為一生成樹。一個圖形的生成樹有許多個，其中邊的總成本最低者為最小成本生成樹。最小成本生成樹不可以有循環(迴路)；最小成本生成樹不必是唯一的。

**Kruskal 演算法：**

假設節點數為 $n$ ，Kruskal 演算法是將各邊先依成本(權重值)的大小由小到大排列，接著從成本(權重值)最低的邊開始加入最小成本生成樹，如果加入的邊會造成循環(迴路)則捨棄不用，直到加了  $n - 1$  個邊為止。舉例說明如何以 Kruskal 演算法得到下圖中的最小成本生成樹：

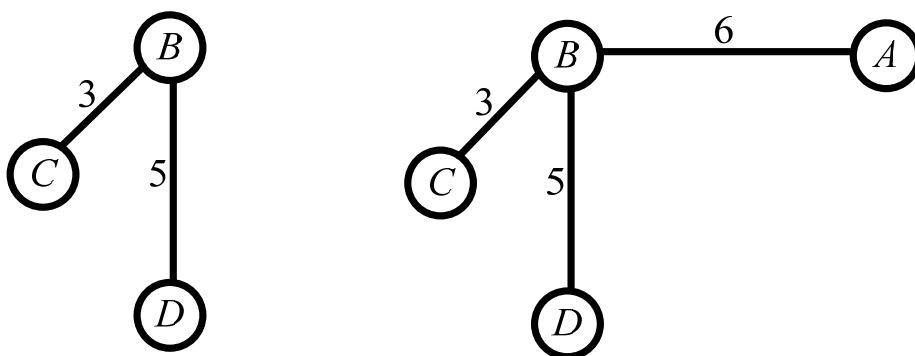


步驟一：將所有邊線的成本(權重值)列出並由小到大排序：

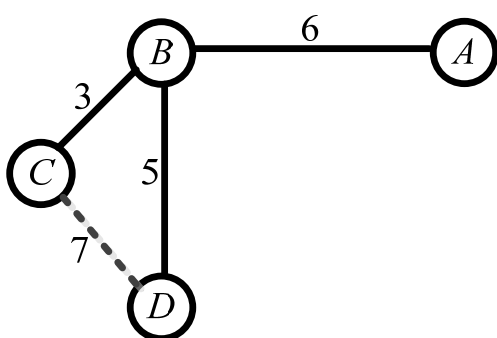
節點	節點	成本(權重值)
B	C	3
B	D	5
A	B	6
C	D	7
B	F	8
A	E	9
D	E	10
D	F	11
A	F	12
E	F	15

步驟二：選擇成本(權重值)最低的一條邊做為加入最小成本生成樹的起點，邊(B,C)->3。

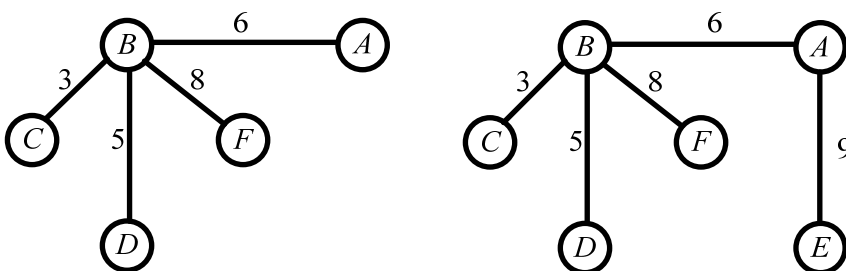
步驟三：依步驟一所建立表格，依序加入邊至最小成本生成樹。



步驟四：邊(C-D) 加入會形成迴路，所以捨棄不用。



重複步驟三和步驟四，直到加了  $n - 1$  個邊為止，完成圖：



最小成本生成樹的值為邊的成本之總合  $3+5+6+8+9=31$

#### 輸入說明：

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

第二列起每一列代表一組測試資料。每組測試資料代表一圖形，內容為邊的資料。每個邊以  $i, j, k$  表示，其中  $i$  和  $j$  為節點的編號，為大寫英文字母(沒有順序)，代表從  $i$  節點和  $j$  節點有邊相連， $k$  為邊的成本(正整數)  $1 \leq N \leq 65535$ ，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， $3 \leq |i, j| \leq 20$ 。

#### 輸出說明：

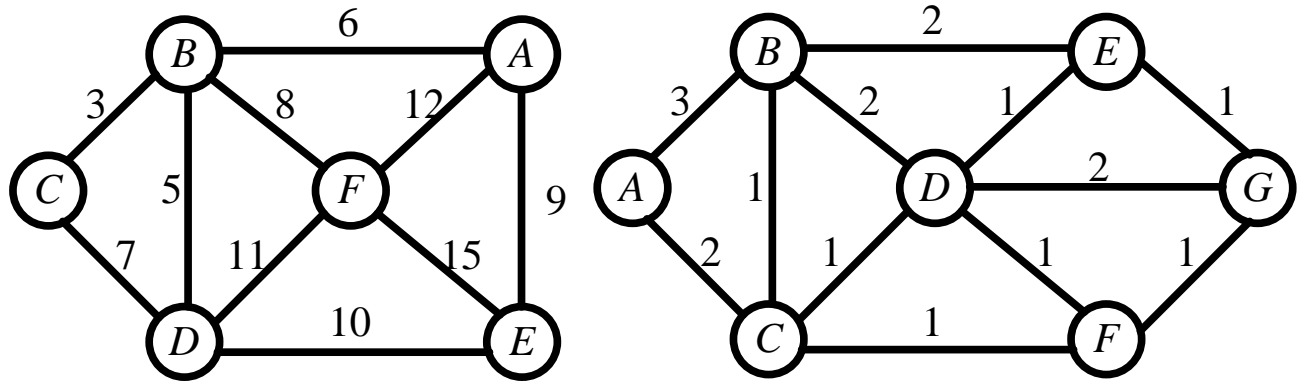
每組測試資料輸出一列。輸出每組測試資料最小成本生成樹的值。



輸入檔案 1 :【檔名 : in1.txt】

2

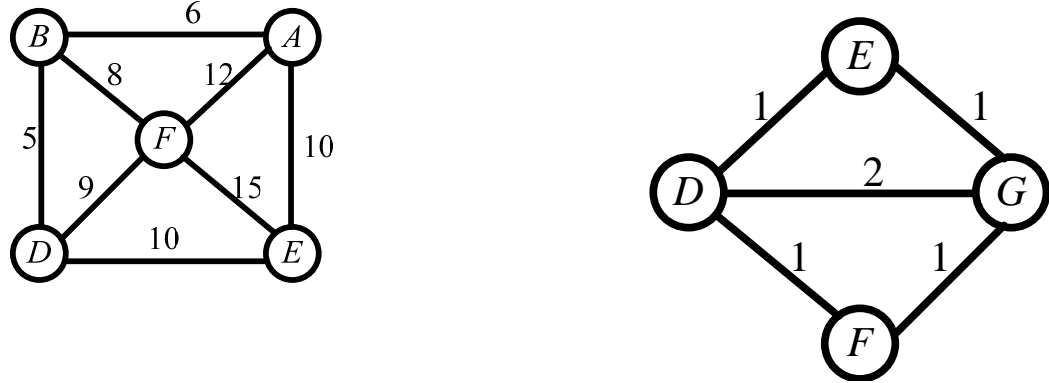
A,B,6 A,E,9 B,C,3 B,D,5 C,D,7 B,F,8 D,E,10 D,F,11 A,F,12 E,F,15  
A,B,3 A,C,2 B,C,1 B,D,2 C,D,1 B,E,2 C,F,1 D,E,1 D,F,1 D,G,2 E,G,1 F,G,1



輸入檔案 2 :【檔名 : in2.txt】

2

B,A,6 B,F,8 B,D,5 D,E,10 D,F,9 A,F,12 A,E,10 E,F,15  
D,E,1 D,G,2 D,F,1 E,G,1 F,G,1



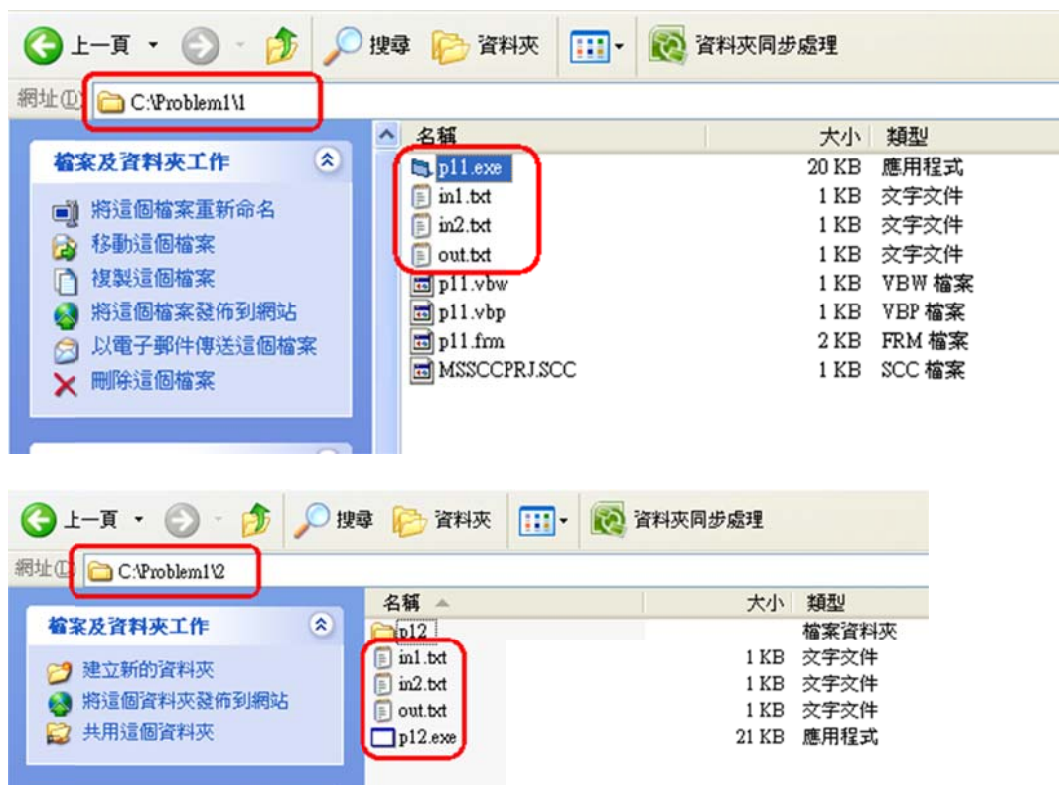
輸出範例 :【檔名 : out.txt】

31  
7  
  
29  
3

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