



全國高級中等學校 101 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：_____ 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。**若結果未依序或不全或未依規定，該子題以零分計算。**

Problem 1：計算字數

子題 1：計算英文單字出現的總次數。(程式執行限制時間: 2 秒) 10 分

請從英文文章或句子中，比對是否出現某英文單字，計算這英文單字在英文文章或句子中出現的總次數，另外計算英文文章或句子中總共有幾個單字。在本題輸入檔案中，而每個單字以逗號(,)、分號(;)、句號(.)或空白()隔開，而空白不限定一個，句子和句子之間可能會出現換行和空行的情況。如 are 在這句子中「Bigtable timestamps are integers.」出現過一次；這個句子，經計算有 4 個單字。大小寫視為相同字元。

輸入說明：

第一列有一個英文單字，計算該英文單字在之後的英文文章或句子中出現的總次數。

第二列以後則是英文文章或句子。文章單字數小於 200 個。

最後一列為 EOF，代表文章或句子的結束，不累加至英文單字。

輸出說明：

根據第一列的英文單字，計算出在第二列起的英文文章或句子中該英文單字出現的總次數，並計算出在這些英文文章或句子中其單字總字數。輸出的英文單字出現總次數和單字總字數，二項輸出資料間以“, ”分開。

輸入檔案 1：【檔名：in1.txt】

the

The villagers of Little Hangleron still called it the Riddle House, even though it had been many years since the Riddle family had lived there. It stood on a hill overlooking the village, some of its windows boarded, tiles missing from its roof, and ivy spreading unchecked over its face.

EOF

輸入檔案 2：【檔名：in2.txt】

are

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

Bigtable timestamps are integers.

EOF

輸出範例：【檔名：out.txt】

4,50

2,23

子題 2：關鍵字排名。(程式執行限制時間: 3 秒) 17 分

在網路搜尋時常會輸入關鍵字，本題事先把這些關鍵字存入檔案 **in1.txt** 和 **in2.txt** 中，請在檔案中，尋找出現最多的三個關鍵字，計算這三個關鍵字出現的次數各為何。在輸入檔中每個關鍵字以空白隔開，而空白不限定一個，中間可能會出現換行和空行的情況。大小寫視為相同字元。

例如在 **in1.txt** 測試資料中，有 9 個關鍵字，其出現次數如下：

google : 10	windows : 3	website : 1
bible : 8	yahoo : 3	analysis : 1
seo : 7	word : 2	quality : 1

則在這組測試資料中關鍵字出現最多的三項為 google、bible 及 seo，由大到小，輸出這三個關鍵字出現次數值為 10,8,7。

例如在 **in2.txt** 測試資料中，有 8 個關鍵字，其出現次數如下：

google : 10	windows : 4	web : 1
linux : 4	yahoo : 4	analysis : 1
seo : 4	time : 2	

則在這組測試資料中關鍵字出現最多的三項為 google 及 linux、seo、windows、yahoo 其中之二，由大到小，輸出這三個關鍵字出現次數值為 10,4,4。若 google 也出現 4 次，則出現最多三項關鍵字之輸出值為 4, 4, 4。

輸入說明：

第一列起是以空白隔開的關鍵字。關鍵字總數小於 200 個。

最後一列為 EOF，代表檔案的結束。

註：檔案中不存在 EOF 的關鍵字，而關鍵字檔案只包括大小寫英文，檔案中包含至少 3 組不同之關鍵字。EOF 只會出現在檔案結尾，表示檔案之結束，不計入關鍵字中。

輸出說明：

由大到小輸出，出現次數最多的前 3 者，其輸出的出現次數間以“, ”分開。

輸入檔案 1：【檔名：in1.txt】

Google Google Google Google Google Bible Bible Bible Bible Google Google Google Google
Google Bible Bible Bible Bible Website Analysis

SEO SEO SEO SEO SEO SEO SEO Quality Windows Windows Windows Yahoo Yahoo Word
Word Yahoo
EOF

輸入檔案 2：【檔名：in2.txt】

Google Google Google Google Google Linux Linux Linux Linux Google Google Google Google
Google Web Analysis SEO SEO SEO SEO Windows Windows Windows Windows Yahoo Yahoo
Time time yahoo Yahoo
EOF

輸出範例：【檔名：out.txt】

10,8,7

10,4,4

Problem 2：資料結構—樹

子題 1：樹的高度。(程式執行限制時間: 2 秒) 10 分

在資料結構中，樹狀結構是可以用來描述有分支的結構，其節點個數是一或一個以上的有限集合。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，都只有唯一的路徑，例如 F 到 A 的路徑為 $F \rightarrow B \rightarrow A$ ，其路徑長度為 3。

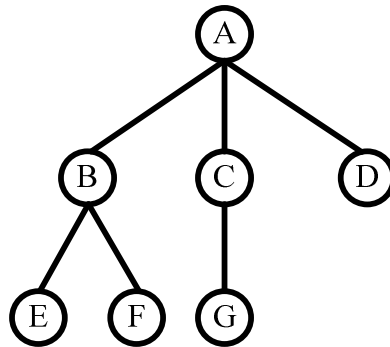


圖 2.1.1

在如圖 2.1.1 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點節點減一，在樹上任意添加一條邊，就會產生迴圈；在樹上任意刪除一條邊，一顆樹就裂成兩棵樹(森林)。沒有迴圈的圖，就是樹或森林。

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)：一個節點子樹的個數稱為其分支度，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 2.1.1，因最大階度階度為 3，則其樹的高度為 3。

在圖 2.1.2 中的圖，則不是樹，因為 B、E、F 三節點形成迴圈。

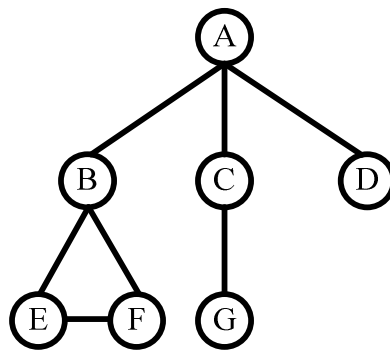


圖 2.1.2

寫一個程式，讀入一樹狀結構的資料，然後回答每組測試資料中，所有節點到根節點路徑長度最大的值。

輸入說明：

第一列的數字 n 代表共有幾組資料要測試，而 n 的值介於 1 和 5 之間。

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，每組測試資料中的第一列值為節點的個數 $m(>=2)$ ，之後的 $m-1$ 列的內容為邊的資料。每組測試資料都是以 0 代表根節點。每個邊以 2 個整數 i,j 表示， $0 \leq i,j \leq 20$ ，其為節點的編號，代表從 i 節點和 j 節點有一個邊相連，節點 j 為節點 i 的父節點，在測試檔中，邊的資料依節點編號順序描述，即 i 的值會 $1,2,\dots, m-1$ 遞增，每組測試資料皆以 0 為根節點。在一行空行之後為下一組的測試資料。

輸出說明：

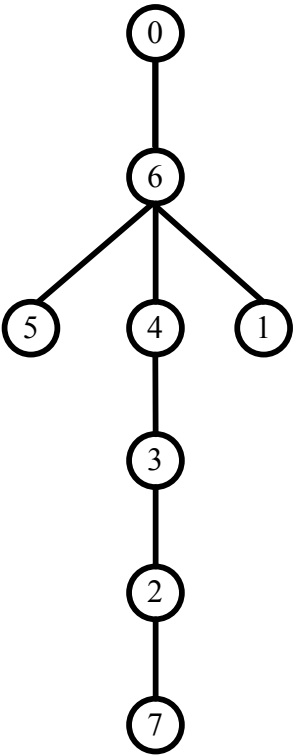
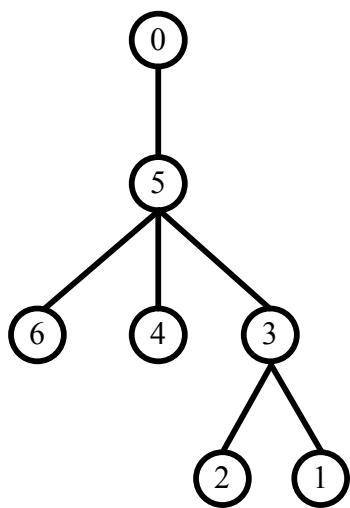
每組測試資料輸出一列，在測試資料中，計算所有節點到根節點 0 的路徑長度，輸出每組測試資料中路徑長度最大的值。

輸入檔案 1：【檔名：in1.txt】

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

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

二組輸入的資料所對應到的樹狀結構。



輸入檔案 2：【檔名：in2.txt】

3
4
1,2
2,3
3,0

4
1,0
2,0
3,0

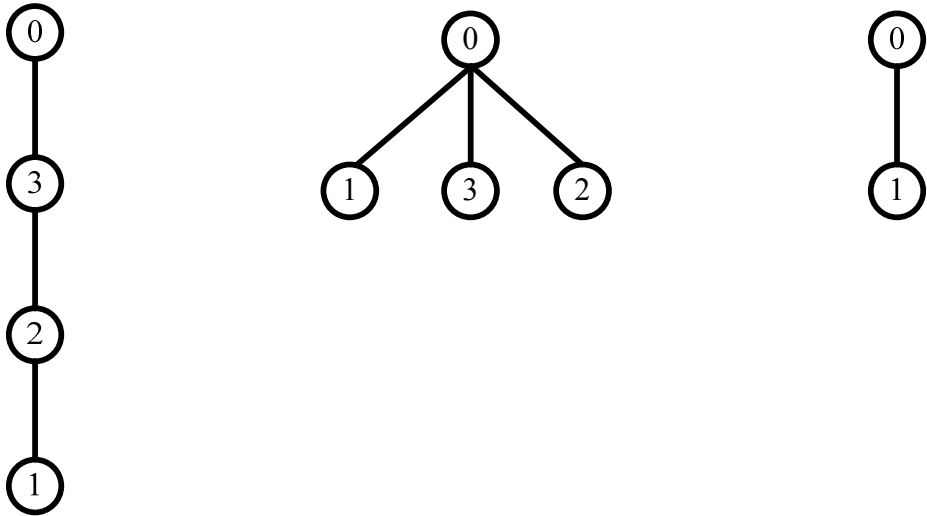
2
1,0

輸出範例：【檔名：out.txt】

4
6

4
2
2

三組輸入的資料所對應到的樹狀結構。



子題 2：樹和樹葉節點。(程式執行限制時間: 3 秒) 14%

寫一個程式，讀入一圖形的資料，然後回答該圖是否為樹，在測試檔中，節點的編號不一定是連續的號碼。樹葉節點為無子節點的節點，若以節點 0 為圖形之根節點，在圖 2.2.1.1 中的節點 1、3、4 及 8 是樹葉節點；在圖 2.2.1.2 中的節點 2、3、4、5 及 10 是樹葉節點；在圖 2.2.2.1 中的節點 4 為樹葉節點。我們假設如果檢測的圖形是樹，其根節點為節點 0，若測試圖是樹則輸出樹葉節點的個數，若該圖不是樹則輸出 F。

輸入說明：

第一列的數字 n 代表有幾組資料要測試，而 n 的值介於 1 和 5 之間。

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， $0 \leq i, j \leq 20$ ，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，同一列中，每個邊的資料以空白()隔開，而空白不限定一個。測試資料中的 0, 0 代表此組輸入資料結束，它不代表一個邊，而是一個稱為節點 0 的節點，如果該圖是一棵樹，節點 0 為圖形之根節點。

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樹。若該圖是樹，則輸出樹葉節點的個數，但根節點不累計為樹葉節點；若該圖不是樹，則輸出 F。(輸出字母為大寫，選手請注意。)

輸入檔案 1：【檔名：in1.txt】

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

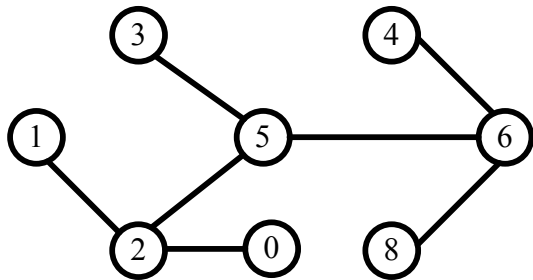


圖 2.2.1.1(in1.txt)

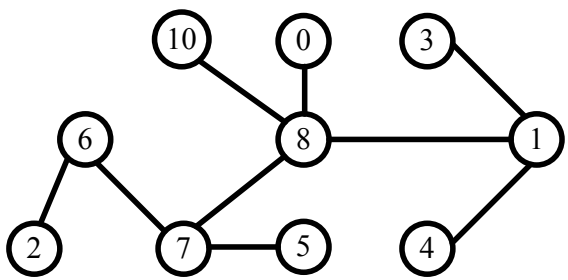


圖 2.2.1.2(in1.txt)

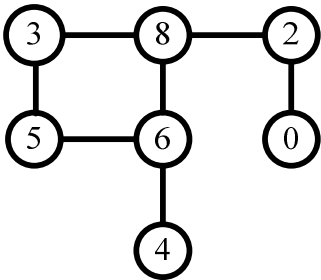


圖 2.2.1.3(in1.txt)

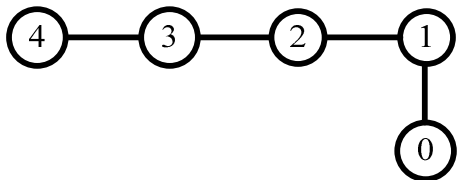


圖 2.2.2.1(in2.txt)

輸入檔案 2：【檔名：in2.txt】

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

輸出範例：【檔名：out.txt】

4
5
F
0
1

1
F
F
F

Problem 3：檢查碼問題

子題 1：身分證。(程式執行限制時間: 2 秒) 9 分

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投票，不限制本人使用，一個有效的身分證字號只能投票一次。請寫一支程式統計(1)有效的身分證字號和(2)有效的身分證字號但重複及(3)無效的身分證字號的個數。

每行的身分證字號將會為三種可能情況之一：

(1)有效的身分證字號：T

(2)有效的身分證字號但重複：O(有效的身分證字號第二次(含)之後再出現都列入重複)

(3)無效的身分證字號：F

分別計算 T、O、F 的個數，並依序輸出其值。

例如在第一組測試資料中，有 8 筆資料：

- | | | |
|------------------|------------------|------------------|
| 1. M123456789(T) | 4. M123456789(O) | 7. A123456789(O) |
| 2. A123456789(T) | 5. M123456789(O) | 8. A223344556(F) |
| 3. A323456783(F) | 6. M123456789(O) | |

例如在第二組測試資料中，有 4 筆資料：

1. R102345678 (T)、 2. A108881111(F) 、 3. A108881111(F) 、 4. B101111111(T)

輸入說明：

第 1 行的數字 n 代表有幾筆測試資料，而 n 的值介於 3 和 10 之間，之後每行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：

共三行，第一行輸出 in1.txt 的執行結果，第二行為空行，第三行輸出 in2.txt 的執行結果，每行輸出 T,O,F 的個數值，對應到每組測試資料中的(1)有效的身分證字號、(2)有效的身分證字號但重複及(3)無效的身分證字號的個數，結果以“, ”分開。

輸入檔案 1：【檔名：in1.txt】

```
8
M123456789
A123456789
A323456783
M123456789
M123456789
M123456789
M123456789
Y123456788
A223344556
```

輸入檔案 2：【檔名：in2.txt】

```
4
R102345678
A108881111
A108881111
B101111111
```

輸出範例：【檔名：out.txt】

```
3,3,2

2,0,2
```

子題 2：編碼。(程式執行限制時間: 2 秒) 17 分

在 ASCII CODE 中，每個字元需要使用 8 bit 來存資料。當檔案只包含 0123456789AB 十二種字元時，可二進制重新編碼以節省空間，假設新編碼如下：

表 3.2.1

二進制	字元
00	A
01	B
100	0
101	1
1100	2
1101	3
11100	4
11101	5
111100	6
111101	7
111110	8
111111	9

例如編碼 10100110001 對應到的字元為 1A2B。在猜數字的遊戲過程中，會選定不重複的數字排列當做答案，再由玩家來猜數字，再算出幾 A 幾 B，其遊戲過程可用表 3.2.1 的編碼紀錄（答案為何，幾 A 幾 B 如何算出，不是這次題目考慮的範圍）。例如玩家猜數字 6789，算出 (0A0B)，則把這過程 6789 (0A0B) 以 111100111101111110111111000010001 編碼，為了讓選手方便對照剛剛的編碼，我們將 6789 (0A0B) 編碼拆解成 111100 (6) 111101 (7) 111110 (8) 111111 (9) 100 (0) 00 (A) 100 (0) 01 (B)；

若玩家猜數字 1253，算出(2A1B)，這過程 1253(2A1B)以表 3.2.1 的方式編碼紀錄為
101 1100 11101 1101 1100 00 101 01

(輸入檔案會省略空白，空白的存在是為了方便讀題)

若玩家猜數字 4321，算出(4B)，這過程 4321 (4B)以表 3.2.1 的方式編碼紀錄為
11100 1101 1100 101 11100 01

若玩家猜數字 5234，算出(3A)，這過程 5234(3A)以表 3.2.1 的方式編碼紀錄為
11101 1100 1101 11100 1101 00

若玩家猜數字 1209，算出(2A)，這過程 1209 (2A)以表 3.2.1 的方式編碼紀錄為
101 1100 100 111111 1100 00

若玩家猜數字 1234，算出(2B)，這過程 1234 (2B)以表 3.2.1 的方式編碼紀錄為
101 1100 1101 11100 1100 01

輸入說明：

第 1 行的數字 n 代表有幾筆資料要測試，而 n 的值介於 1 和 5 之間，之後每行為 0 和 1 所組成的編碼字串，字串長度 ≤ 34 ，對應到一次猜數字的遊戲過程。在測試檔案中，每個編碼字串均可正確的對應到編碼表中的編碼。

輸出說明：

從第 1 行起每行將輸入之編碼字串，轉成玩家猜的數字及其幾 A 幾 B 的結果。（輸出英文字均為大寫，選手請注意。數字和其幾 A 幾 B 的結果以“,”分開。）

輸入檔案 1：【檔名：in1.txt】

```
4
1111001111011111101111111000010001
101110011101110111000010101
11100110111001011110001
111011100110111100110100
```

輸入檔案 2：【檔名：in2.txt】

```
2
1011100100111111110000
1011100110111100110001
```

輸出範例：【檔名：out.txt】

```
6789,0A0B
1253,2A1B
4321,4B
5234,3A

1209,2A
1234,2B
```

Problem 4：其他

子題 1：進制轉換 (程式執行限制時間: 2 秒) 10 分

我們日常生活用的進制為十進制，數字的組合由 0~9 構成，而二進制的數字組合只有 0 和 1，十進制超過 9 則左邊的位數便加 1，而二進制超過 1 則左邊的位數便加 1。

為什麼要使用二進制呢？因為電腦只認得 0 與 1 的信號。十六進制或八進制的使用可將一長串的二進制，換成十六進制或八進制，較容易閱讀，且二進制轉換成十六進制或八進制是十分便利的。

下表列出十進制 0~15 的二進制、十進制與十六進制的對照：

二進制	十進制	十六進制
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

下表列出十進制 0~7 的二進制、十進制與八進制的對照：

二進制	十進制	八進制
000	0	0
001	1	1
010	2	2
011	3	3
100	4	4
101	5	5
110	6	6
111	7	7

輸入說明：

第 1 行的數字 n 代表有幾筆資料要測試，而 n 的值介於 1 和 5 之間，接下來是要轉換的二進制字串(每筆二進制字串資料長度 ≤ 16)。

輸出說明：

將輸入的二進制字串轉換成十六進制和八進制，結果以“, ”分開。二進制字串轉換成十六進制時，由右而左，每四個二進制字串為一組，當最左邊的一組不足四個二進制字串時，亦要補足四個二進制字串再計算；當二進制字串轉換成八進制時，由右而左，每三個二進制字串為一組，當最左邊的一組不足三個二進制字串時，亦要補足三個二進制字串再計算。若最左邊的一組二進制字串轉換成十六進制或八進制，結果為 0 時，亦要輸出。

輸入範例：【檔名：in1.txt】

```
3
010010011
00011110001
11001100
```

輸入範例：【檔名：in2.txt】

```
2
00110111
11011110
```

輸出範例：【檔名：out.txt】

```
093,223
0F1,0361
CC,314
```

```
37,067
DE,336
```


子題 2：撲克牌遊戲（程式執行限制時間: 2 秒） 13 分

許多人常喜歡玩撲克牌廿一點的遊戲，一副牌共有 52 張牌，有四種花色：黑桃、紅桃、方塊、和梅花。在廿一點的玩法中，A 可作 1 點或 11 點，而 2-10 則為該牌之點數，另外 J、Q、K 皆為 10 點。擁有最高點數且其點數等於或低於 21 點的玩家獲勝，超過 21 點的玩家稱為爆牌。若玩家會因 A 以 11 點計算時而爆牌，則該牌 A 必須以 1 點計算；若玩家拿到 A 算 11 點沒爆牌，則該牌 A 必須以 11 點計算。本程式只考慮算點數，不考慮其他的下注分牌動作。在測試檔案中，每位玩家只會分到 2 張牌或 3 張牌。下表將 52 張牌分別對應到數字 1~52，在測試檔案中，將以數字代表某張牌。

點數 花色	A	2	3	4	5	6	7	8	9	10	J	Q	K
黑桃	1	2	3	4	5	6	7	8	9	10	11	12	13
紅桃	14	15	16	17	18	19	20	21	22	23	24	25	26
方塊	27	28	29	30	31	32	33	34	35	36	37	38	39
梅花	40	41	42	43	44	45	46	47	48	49	50	51	52

輸入說明：

每個輸入資料含多個玩家取得的撲克牌資料，檔案 in1.txt 和 in2.txt 分別各使用一副牌，第 1 行的數字 n 代表有幾筆玩家資料要測試，而 n 的值介於 1 和 5 之間，之後每行代表每個玩家拿到的撲克牌，可能有 2 張牌或 3 張牌，每張牌以空白()隔開，而空白不限定一個。每張牌以一個數字(1~52)代表，例如紅桃 5 以 18 代表。

輸出說明：

按照每個玩家手上的 2 張牌或 3 張牌，計算出玩 21 點時的點數，並依序輸出。爆牌則輸出 F，低於 21 點(含)則輸出每個玩家的總點數。

輸入範例 1：【檔名：in1.txt】

5

3 44

6 12 1

26 25 2

14 27

40 43

輸入範例 2：【檔名：in2.txt】

3

15 18 2

14 21

1 13 26

輸出範例：【檔名：out.txt】

8

17

F

12

15

9

19

21