

282.家教推薦中心(2 分)

Problem Description

為協助同學找到適合的家教工作，學生活動中心擬成立網路家教推薦中心來撮合本校想要當家教的同學和校外需要課業輔導的學生。假設本校登記為家教的同學需於期初填入下列資料：家教姓名、擬教科目、擬教時間、勝任程度、期望最低鐘點費。需要輔導學生的資料為：學生姓名、輔導科目、時間、願付鐘點費。

各欄位的資料格式說明如下：

- 家教姓名和學生姓名均為中文，均為 6 個 byte。
- 擬教科目和輔導科目以大寫英文字母代表。
- 勝任程度為家教推薦中心根據登記者的科系、歷來表現評比所記錄的。以 A、B、C、D 依序表示勝任程度。A 乃最為勝任。
- 時間以 1、2、3、4、5、6 表示週一到週六。
- 鐘點費以整數表示。

家教推薦中心將依據雙方的資料進行撮合。撮合的原則之優先順序為

1. 擬教科目與輔導科目必須吻合，兩者的時間亦必須吻合。
2. 鐘點費的支給以輔導學生願意支付的鐘點費為準。
3. 勝任程度較高的家教較優先分配到學生。若有家教或多個學生均互相符合條件時，勝任程度較高的家教分配到較高鐘點費的學生。
4. 若科目與時間吻合，但是鐘點費未達家教期望最低鐘點費，則不予分配。
5. 若有多個登記者同時符合以上條件，而無法排出順序時，則以先登記者優先。

Input File Format

第一列有兩個正整數 n 與 m ，其中 n 代表登記為家教的人數， m 為登記輔導的學生人數。第二列開始的 n 列為依登記先後順序排列的家教資料，每列家教資料有家教姓名、擬教科目、擬教時間、勝任程度、期望最低鐘點費。接著是 m 列的輔導學生資料，每列資料有學生姓名、輔導科目、時間、願付鐘點費。

上述輸入資料，若每列超過一個資料，則資料間以一或多個空白分隔。此外，上述輸入資料保證格式正確。注意： $m < 100$ ， $n < 100$ 。

Output Format

印出所有配對，每列印一個配對，包含家教姓名、學生姓名以、科目、時間、鐘點費。資料間以一個空白分隔。

Example

Sample Input:	Sample Output:
6 7 陳俊恆 A 135 A 500 嚴家慶 B 246 B 600 蔣國安 A 135 B 600 薛乃仁 B 246 A 600 林姿寶 C 135 B 400 姚伊宣 C 135 A 500 王大同 A 135 700 李大華 B 135 400 陳小東 C 135 600 蕭中平 A 246 500 戴中仁 B 135 500 齊小玲 C 246 450 范小宣 A 135 500	陳俊恆 王大同 A 135 700 姚伊宣 陳小東 C 135 600

283.資料壓縮(2 分)

Problem Description

資料壓縮 (data compression) 是減少資料儲存空間或資料傳輸量的一種方式，其應用相當廣泛。資料壓縮可分成兩大種類，有失真壓縮與無失真壓縮。無失真壓縮方法所壓縮的資料經過解壓縮 (uncompress) 後，還原的資料與壓縮前完全一樣，常用於檔案壓縮。有失真壓縮方法的資料解壓縮 (uncompress) 後，還原的資料與壓縮前不完全一樣，有些許差異，但影響不大，常用於多媒體資料。目前已有許多方法，可對影像資料進行壓縮。一張影像為一個二維陣列的資料，每個元素為一個像素 (pixel)，一個像素以 8 bits 表示(範圍為 0 至 255)。此處的座標系統定義左上角為(0,0)，其右側一點為(0,1)，下方一點為(1,0)。若有一個 3 4 (高為 3，寬為 4) 的影像，則其座標值如下：

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3)
(2,0)	(2,1)	(2,2)	(2,3)

如果一張影像有許多像素為 0，可用下面的方法來進行壓縮。例如一張 7 7 影像如下：

0	0	0	0	0	0	0
0	0	34	0	0	0	0
13	6	12	0	0	0	0
0	0	37	5	76	0	0
0	0	68	4	0	0	0
0	0	0	2	0	0	0
0	0	0	0	0	0	0

假設以座標(3,2)為起點，其像素值為 37。我們可將其周圍連接的像素，從左上角起，依順時針方向，記載其為 0 或非 0。若為 0，則記為 Z，若為非 0，則記為 Y。例如 37 周圍 8 個像素特徵為 YYZYZZ，另外再記錄非 0 像素的像素值，其為 6,12,5,4,68。接著再由此 8 個像素的第一個像素往外擴充，但是，像素本身為 0 者，則不再擴充，如第三個像素；此外，擴充時，若其周圍連接的像素以前曾經列出過，則跳過不予列出，例如將 6 擴充後，其像素特徵為 ZZYZY，像素值為 34,13，並將像素特徵與像素值分別跟目前已經找到者相接起來。然後，再將 8 個像素中，下一個為 Y 者（即 12），加以擴充。另外，如果擴充後的像素已經超出影像範圍，則跳過不予列出。依照此方法，即可將影像中相連的部分全部列出。例如，上述影像全部的像素特徵為（其中第一個 Y 為代表起點像素）：

Y,YYZYZZ,ZZYZY,Z,ZYZ,ZYZ,Z,ZZZ,ZZZ,ZZZ

其像素值為

37,6,12,5,4,68,34,13, ,76,2, , , ,

請注意，本題所給的影像一定全部連在一起，而且起始點像素不為 0。

Input File Format

第一列有一個正整數，代表測試資料之組數 n ， $n < 10$ 。其後有 n 組測試資料，每一組的第一列為一個英文字母 C 或 U。若為 C 代表該組資料為影像資料，需要將其壓縮；其後一列有四個非零整數 m, k, i, j ，分別代表影像高度為 m ， $m < 50$ ，寬度為 k ， $k < 50$ ，起始點為 (i, j) ；之後有 m 列代表一個影像，每列有 k 個整數（範圍為 0 至 255）。若第一列的英文字母為 U，代表該組資料為壓縮過的資料，需要解壓縮；其後一列有六個非零整數 m, k, i, j, p, q ，分別代表影像高度為 m ， $m < 50$ ，寬度為 k ， $k < 50$ ，起始點為 (i, j) ，像素特徵（YZ 系列）共有 p 個，像素值共有 q 個；之後有多列為像素特徵（每列至多 20 個像素特徵，像素特徵太長時，可置於第二列，第三列...），再之後有多列為像素值（每列至多 20 個像素值，像素特徵太長時，可置於第二列，第三列...）。

上述輸入資料，若每列超過一個資料，則資料間以一或多個空白分隔。但是，像素特徵則為連續資料，其間沒有空白。此外，上述輸入資料保證格式正確，其邏輯性亦完全正確。

Output Format

若為壓縮動作，則印出資料與「輸入格式」的第二種格式一樣；若為解壓縮動作，則印出資料與「輸入格式」的第一種格式一樣。

Example

Sample Input:	Sample Output:
2	7 7 3 2 31 10
C	YYYZYZZZZYZZZYZZY
7 7 3 2	ZZZZZZZZZZ
0 0 0 0 0 0	37 6 12 5 4 68 34 13 76 2
0 0 34 0 0 0	7 7 3 2
13 6 12 0 0 0	0 0 0 0 0 0
0 0 37 5 76 0 0	0 0 34 0 0 0
0 0 68 4 0 0 0	13 6 12 0 0 0
0 0 0 2 0 0 0	0 0 37 5 76 0 0
0 0 0 0 0 0 0	0 0 68 4 0 0 0
U	0 0 0 2 0 0 0
7 7 3 2 31 10	0 0 0 0 0 0
YYYZYZZZZYZZZYZZY	
ZZZZZZZZZZ	
37 6 12 5 4 68 34 13 76 2	

284.交通號誌設定(1 分)

Problem Description

在市區的交通網路中，於交叉路口設置交通號誌(紅綠燈) 是不可避免的事。在相鄰兩個路口，綠燈亮的時間若互相配合得當，則行車順暢；反之，在一個路口通過了綠燈，到下一個路口卻是紅燈，則需要等待。

考慮一條直路上的交通號誌時間設定問題，如下圖：

現在，有一位駕駛人於時間 0 要從 A 出發，最後的目的點是 E。路段上的數字表示走完該路段所需之秒數，例如 A 到 B 需 2 秒，B 到 C 需 6 秒。假設交通號誌只有紅燈與綠燈，其週期為 T 秒，即每個路口紅燈亮 $T/2$ 秒，綠燈亮 $T/2$ 秒，持續循環。上圖中，各點的數字為該點綠燈開始亮的時間，其介於 0 與 $T-1$ 之間。假設 $T=10$ ，到達 E 點之時間可如下計算：

A 點需等 3 秒(因為 0 出發，但 3 才變為綠燈)；到 B 點為時間 5，而 B 點此時為綠燈 (2~6 為綠燈，7~11 為紅燈)，故不需等待。到 C 點為時間 11，而 C 點之 5~9 及 15~19 為綠燈，故需等待 4 秒，當時間 15 才能再從 C 出發；到 D 點為時間 18，而 D 點綠燈時間為 9~13 及 19~23，故需等待 1 秒。D 至 E 之長度為 5，故於時間 24 抵達 E(因已到達 E，故不必計算 E 之等待時間)。所以駕駛人於時間 0 從 A 出發，在時間 24 會抵達 E 點。如果有另一位駕駛人於時間 0 從 E 走向 A，其抵達 A 的時間亦可算出。注意：如果某一點目前為綠燈，則雙向均可通行，若為紅燈，則雙向均不可通行。如果將綠燈亮的時間加以變更，如下：

我們可以算出從 A 到 E 的總時間為 16，從 E 到 A 的總時間亦為 16。此設定亦是最好的設定，因為途中均無需等待紅燈。本題即是要找出設定綠燈亮的時間，使得兩個方向從時間 0 出發（第一點為起點），於抵達目的點（最後一點）所需要的時間相同或者最多相差 1。在符合條件的眾多設定中，找出兩者所花時間合計為最小者。若仍有許多組設定方式，則任何一種均是本題答案。

Input File Format

每一組測試資料有四列；其中第一列為路段個數 n ， $n < 15$ ；第二列有一個非負整數，為週期 T (一定為正偶數， $0 < T < 10000$)；第三列為 n 個路段之長度 (均為整數，且小於 10000)。一組測試資料結束之後，可能有下一組測試資料，若接下來的路段個數 n 為 0，則表示所有測試資料結束。

Output Format

每一組測試資料印出兩列，第一列印出兩個方向所需時間總和，第二列印出 $n+1$ 個點（含首尾兩點）綠燈開始亮的時間(均介於 0 與 $T-1$ 之間)。

Example

Sample Input:	Sample Output:
4	32
10	0 2 6 1 0
2 6 3 5	1801
2	0 201
1000	
700 200	
0	

285.商店促銷(2 分)

Problem Description

為達到有效的促銷，商店擬從結帳櫃臺的系統所蒐集的客戶交易資料來找出適合一起促銷的產品。每一次結帳視為一個交易，而每一交易包括一個以上的產品項目。例如，{A, B, C} 表示在一次結帳時買了 A、B、C 三種產品。而促銷的策略為，如果已知過去有哪些產品項目（稱為集合 L）一起被購買的機會超過某一比例，則對某些客戶而言，如果曾經購買過這些項目的部份集合（稱為集合 M，而且 $M \subseteq L$ ），那麼向這些客戶促銷產品項目在集合 L 和 M 的差集(即 $L-M$)的成效將可大增。為確定達到某個顯著水準，一般而言，我們可以指定一個支持水準(support rate)：所選定的項目集合佔所有交易筆數的比率。例如，有如下的四筆交易：

交易 1：A, B, C

交易 2：A

交易 3：C,D,E

交易 4：A, B

A 產品被購買的支持水準為 75% (亦即 $p(A)=3/4$)，而 A 產品和 B 產品一起被購買的支持水準為 50% (亦即 $p(A,B)=1/2$)。買 A 的交易，同時也買 B 時，佔 A 被買的比率之 $2/3$ ；買 B 的交易，同時也都買了 A。也就是 $p(A,B)/p(A)=2/3$ ； $p(A,B)/p(B)=1$ ，我們稱此為信心水準(confidence)。如果我們欲找出支持水準大於或等於 50%，的促銷法則，則我們會找到買 A(支持水準大於或等於 50%)後，再買 B(信心水準大於或等於 60%)的法則，記為 $A \rightarrow B$ 。

舉另一個例子，假設我們收集了一段時間的交易資料後，我們指定的支持水準為大於或等於 50%，信心水準為大於或等於 60%。如果我們已經算出同時購買 A、B、C 三項產品大於或等於支持水準 50%，然後我們可以根據交易資料，算出其中兩個產品購買後，會再買第三個產品的信心水準。例如：我們算出 $p(A,B,C)/p(A,B)=1/2$ ； $p(A,B,C)/p(A,C)=1/2$ ； $p(A,B,C)/p(BC)=2/3$ ，則我們輸出的促銷法則為 $BC \rightarrow A$ 。

假設我們有 10 種產品項目(從 A 到 J)，在我們收集了一段時間的交易資料後，我們欲根據所指定的支持水準和信心水準，列出所有可以放在一起促銷的產品。

Input File Format

第一列為交易資料筆數 n。第二列為支持水準和信心水準(以百分比表示)。第三列開始為交易資料，(每一筆交易資料的產品項目已經按字母排序)。

上述輸入資料，若每列超過一個資料，則資料間以一或多個空白分隔。此外，上述輸入資料保證格式正確。注意：n<1000。

Output Format

輸出所有可能的促銷項目，每列印出一個促銷法則。

Example

Sample Input:	Sample Output:
10	A -> B
50 70	B -> A
A B C	B -> C
A B D	D -> C
A B C	A B -> C
A B C E	A C -> B
C D E F	B C -> A
C E F	
B C D F G	
A B C D E F G	
C D E G J	
A B C D F G J	

286.打賭遊戲(2 分)

問題描述：

三個箱子中有一個放了巨額獎金，另兩個是空的。參賽者選擇一個箱子後，莊家打開一個空的箱子，問參賽者要不要改變選擇換到另一個箱子。此時參賽者有三種可能的策略：

1. 維持原來之選擇
2. 換到另一個箱子
3. 隨機（ random ）決定要不要換

最後莊家公佈答案，決定勝負。請寫一程式完成這個遊戲。

提示：

遊戲開始前必須顯示三種策略之代號(參考 Sample Output)。程式必先隨機決定巨額獎金放置之箱子，但尚不能公佈，參賽者此時決定其選擇之箱子，接著莊家打開一個空的箱子之後，參賽者再決定採用何種策略，此時顯示參賽者最後選擇之箱子，然後公佈答案及猜中與否。

輸入說明：

輸入選擇之箱子，等莊家揭開一空箱子後，再輸入所採用之策略。

輸出說明：

遊戲開始前先印出三種策略之代號及遊戲說明，印出參賽者選擇之箱子、一空箱號碼及參賽者最後之選擇，公佈巨額獎金所在位置並印出成功與否。

範例：

Sample Input:	Sample Output:
1 1	=Game Role= 1.Choose one box that you think with bonus. 2.Wait for the banker announced an empty box then choose a strategy code. Strategy Code: 1.Keep the original choice. 2.Change to the other box. 3.Random selection. ***** Player select:1 Empty box:2 Final select:1

	>the Bonus is in the box 3 >>FAILED
--	--

287.即時工作排程(2 分)

問題描述：

即時工作有到達時間、執行期間、完成期限 (deadline)、以及優先次序 (priority) 等屬性。例如，某一即時工作 J 可能於時間 T 到達系統，在 T_d 之前必須完成此工作，其執行所需要花費的時間為 T_e 。排程策略 (scheduling policy) 用來排列出即時工作執行順序。今定義排程策略 $P+EDF$ 。令優先次序值等於 1 為最高優先執行。EDF 代表 earliest deadline first。本策略對於目前已經到達的工作，令優先次序較高者有優先執行的權利；若兩工作優先次序相同，則比較兩者的完成期限，完成期限較早者有優先執行權。一個工作完成後，再依相同策略，選擇下一個執行的工作（可能又有新到達的工作，因此，目前已經到達的工作數量可能增加）。

輸入說明：

第一列有一個正整數，代表即時工作之個數 n ，請注意 $n \leq 10$ 。其後有 n 列，每一列代表一個即時工作。每一列之資料依序為工作 ID、工作到達時間、執行期間、完成期限、及其優先次序。各項資料之間，以一個空白分隔。工作 ID 為英文字母 (不超過 10 個字元)，各時間均為正整數，優先次序為一個正整數 (≤ 10)。注意：本題輸入之資料，保證每一個工作均可在其完成期限之前完成，故毋需考慮無法完成工作之情形。另外， n 列工作資料並不一定按工作到達時間的順序輸入。

輸出說明：

印出一列，依排程策略所排列出的工作執行順序 (列出工作 ID 執行順序)。各項資料之間以一個空白分隔。

範例：

Sample Input:	Sample Output:
4 A1 3 3 30 8 A2 6 4 25 4 A3 4 5 30 4 A4 8 3 30 1	A1 A2 A4 A3

288. The Robots(1 分)

問題描述：

寫一個程式預測兩個機器人的命運。 假設我們有兩個機器人，在 M (水平) \times N (垂直)單位的方格上行動。兩個機器人在一單位的時間內都會移動一格。

第一個機器人有 $F1$ 單位的燃料，而第二個有 $F2$ 單位，如果一個機器人用光了燃料，他會停在那一格上。移動一格需要一單位的燃料。在一開始的 $N1$ 時間內第一個機器人 $R1$ 會往北方移動，然後接下來的 $E1$ 的時間會往東；第一個機器人會重複這兩個步驟直到耗光燃料。第二個機器人的行動稍微不一樣，他在一開始的 $E2$ 時間會先往東移動，然後在接下來的 $N2$ 時間內往北移動，並且重複這兩個步驟直到耗盡燃料。

假如有某個機器人移動超出地圖範圍，他會在地圖上的另一端重新出現。 例如，如果 $M = 7, N = 6$ ，機器人在 $(5, 5)$ 並且往北走，他會重新在 $(5, 0)$ 出現。另外，如果兩個機器人移動到同一個格子上，兩個就會撞在一起而爆炸。現在給定第一個機器人的起點 $(X1, Y1)$ ，第二個機器人的起點 $(X2, Y2)$ ，兩個機器人的燃料量 $(F1 \text{ and } F2)$ ，請判斷這兩個人機器人會不會爆炸。

輸入說明： 輸入只有一行，含有這些整數 $M, N, X1, Y1, E1, N1, F1, X2, Y2, E2, N2, F2$ ，且依照下列限制：

- $10000 > M, N > 0$
- $0 \leq X1, X2 < M$
- $0 \leq Y1, Y2 < N$
- $(X1, Y1)$ is not $(X2, Y2)$
- $N1, E1, N2, E2 > 0$
- $0 \leq F1, F2 \leq 10000$

輸出說明： 輸出有兩種可能。如果機器人爆炸了，請輸出一行 "robots explode at time T "， T 是爆炸的時間。

否則請輸出 "robots will not explode"。

範例：

Sample Input:	Sample Output:
7 6 2 0 9 2 100 3 5 2 7 100	robots explode at time 5
7 6 2 0 9 2 6 3 5 2 7 0	robots will not explode

289. The Robots(1 分)

問題描述：

寫一個程式預測兩個機器人的命運。 假設我們有兩個機器人，在 M (水平) \times N (垂直)單位的方格上行動。兩個機器人在一單位的時間內都會移動一格。

第一個機器人有 $F1$ 單位的燃料，而第二個有 $F2$ 單位，如果一個機器人用光了燃料，他會停在那一格上。移動一格需要一單位的燃料。在一開始的 $N1$ 時間內第一個機器人 $R1$ 會往北方移動，然後接下來的 $E1$ 的時間會往東；第一個機器人會重複這兩個步驟直到耗光燃料。第二個機器人的行動稍微不一樣，他在一開始的 $E2$ 時間會先往東移動，然後在接下來的 $N2$ 時間內往北移動，並且重複這兩個步驟直到耗盡燃料。

假如有某個機器人移動超出地圖範圍，他會在地圖上的另一端重新出現。 例如，如果 $M = 7, N = 6$ ，機器人在 $(5, 5)$ 並且往北走，他會重新在 $(5, 0)$ 出現。另外，如果兩個機器人移動到同一個格子上，兩個就會撞在一起而爆炸。現在給定第一個機器人的起點 $(X1, Y1)$ ，第二個機器人的起點 $(X2, Y2)$ ，兩個機器人的燃料量 $(F1 \text{ and } F2)$ ，請判斷這兩個人機器人會不會爆炸。

輸入說明： 輸入只有一行，含有這些整數 $M, N, X1, Y1, E1, N1, F1, X2, Y2, E2, N2, F2$ ，且依照下列限制：

- $10000 > M, N > 0$
- $0 \leq X1, X2 < M$
- $0 \leq Y1, Y2 < N$
- $(X1, Y1)$ is not $(X2, Y2)$
- $N1, E1, N2, E2 > 0$
- $0 \leq F1, F2 \leq 10000$

輸出說明： 輸出有兩種可能。如果機器人爆炸了，請輸出一行 "robots explode at time T "， T 是爆炸的時間。

否則請輸出 "robots will not explode"。

範例：

Sample Input:	Sample Output:
7 6 2 0 9 2 100 3 5 2 7 100	robots explode at time 5
7 6 2 0 9 2 6 3 5 2 7 0	robots will not explode

290. The Typhoon(1 分)

問題描述：

請寫一個程式來預測一個颱風的移動路徑。假設有一個半徑 R 的颱風中心在 (x, y) 。現在我們想計算這個颱風在 16 個方向上的"阻力"。計算阻力的方向請依照下圖所示。阻力的公式為： $r(x, y) = ax^2 + bxy + cy^2 + dx + ey + f$ 。每次在找到阻力最小的方向之後(在這 16 個方向之中)，這個颱風會按這個方向以固定速度 g 移動一小段時間(dt)。如果有超過一個方向有相同並且最小的阻力，颱風會選擇索引值最小方向(如下圖中的索引)。當颱風移動到新的位置之後，他會重新計算阻力並且移動。這個模擬必須持續一段時間 T 。

現在給定颱風中心的初始位置 (x, y) ，半徑 R ，阻力公式的係數 (a, b, c, d, e, f) ，固定的速度 g ，時間間隔 dt ，以及模擬的總時間 T 。請模擬颱風的行動，並在每個時間點印出颱風中心的位置

輸入說明： 輸入含有下列雙精度浮點數 -- $x, y, R, a, b, c, d, e, f, g, dt$, and T .

輸出說明： 輸出有 $T/dt + 1$ 行。第一行是模擬時間為 0 時颱風的初始位置 (x, y) 。第二行是在 dt 時間時颱風的位置，第三行是在 $2dt$ 時間時的位置，以此類推。 x 和 y 以空白格開，並且以 $\%.3lf$ 的格式輸出。請記得用 0.000 取代 -0.000。

範例：

Sample Input:	Sample Output:
0.000 0.000 1.000	0.000 0.000
1.000 2.000 3.000	
4.000 5.000 0.000	-0.462 -0.191
1.000 0.500 3.000	
	-0.962 -0.191

	-1.462 -0.191
	-1.924 0.000
	-1.462 -0.191
	-1.924 0.000

291. 霓虹燈泡(1 分)

問題描述：

一個大型霓虹燈由 n 個燈泡所組成，燈泡編號 1 至 n 。工程師設計霓虹燈的閃爍規則共分 n 個階段，第 i 個階段，編號是 i 的倍數的燈泡暗亮反相，也就是說燈泡本來是暗的變成亮的、亮的變成暗的。假設剛開始的時候所有的燈泡都是暗的。請問最後總共有幾個燈泡是亮的？

輸入說明：

程式的輸入包含兩行數字，第一行包含一個正整數 k ， $1 \leq k \leq 10$ ，代表第二行有 k 個測試資料 n_1, n_2, \dots, n_k ， $2 \leq n_i \leq 100000$ ，而此 k 個正整數間以空格隔開。

輸出說明：

輸出 k 列答案，針對每一個測試資料 n_i ，輸出一個答案 m_i 。

範例：

Sample Input:	Sample Output:
3	2
5 43 100	6
	10

292. 先至先服務機制(FCFS)問題 I(1 分)

問題敘述

模擬有 n 個處理器的 FCFS 排班機制操作系統環境並且計算平均執行時間。我們假設其他的資源相當充分足夠執行，Queue 的長度無限大，並且我們忽略處理器之間的溝通時間。

Input Format

Enter an input file ****.txt (include processor number, arrival time and execution time).

Output Format

Print the average processing time .

Sample

	Sample Input	Sample Output
1	Number of Processor : 10 Arrival Times : 10 20 30 40 50 60 70 80 90 100 Execution Times : 20 20 20 20 20 20 20 20 20 20	Average Processing Time : 12
2	Number of Processor : 20 Arrival Times : 10 20 40 50 70 100 100 120 130 140 150 160 180 220 310 320 350 400 420 430 Execution Times : 10 50 30 70 60 10 20 70 60 50 20 60 90 100 20 30 50 70 70 50 90	Average Processing Time : 12

3	Number of Processor : 10	Average Processing Time : 28
	Arrival Times :	
	10 25 30 45 50 65 70 85 95 100	
	Execution Times :	
	20 30 40 20 30 10 40 50 70 90	

293. 先至先服務機制(FCFS)問題 II(1 分)

問題敘述

模擬有 n 個處理器的 FCFS 排班機制操作系統環境並且計算平均執行時間。我們假設其他的資源相當充分足夠執行，Queue 的長度無限大，並且我們忽略處理器之間的溝通時間。

Input Format

Enter an input file ****.txt (include processor number, arrival time and execution time).

Output Format

Print the average wait time .

Sample

	Sample Input	Sample Output
1	Number of Processor : 10 Arrival Times : 10 20 30 40 50 60 70 80 90 100 Execution Times : 20 21 22 23 24 25 26 27 28 29	Average Wait Time : 5
2	Number of Processor : 20 Arrival Times : 10 20 40 50 70 100 100 120 130 140 150 160 180 220 310 320 350 400 420 430 Execution Times : 10 50 30 70 60 100 20 30 50 70 50 90 10 20 70 60 50 20 60 90	Average Wait Time : 47

294.判斷四條直線(3 分)

問題描述：

給予一個 5x5 的數字矩陣，內容為由 1 到 25 沒有重複的數字，稱為「目標矩陣」。

另外給予一串由 1 到 25 沒有重複的 25 個數字序列，代表選取的序列。請寫出程式判斷在第二個數字序列中由第一個數字開始，

各數字在「目標矩陣」中出現的位置標示後，在第二個數字序列中第幾個數字會在矩陣中足夠形成至少四條

至少由四個數字形成的直線或斜線。若是選取序列在「目標矩陣」中有五個數字形成一條直線時，則只可計算為一條直線。

輸入說明：

輸入分為兩部份，第一列是由英文逗號分隔開的由 1 到 25 中沒有重複的 25 個數字，每 5 個數字形成「目標矩陣」的一列，

由「目標矩陣」的第一列開始，這 25 個數字剛好表示 5 列。

第二列部份也是由英文逗號分隔開的由 1 到 25 中沒有重複的 25 個數字，代表選取的序列元素。

輸出說明：

第一列為在選取序列中的第幾個數字，其前面的所有數字在「目標矩陣」中的位置足夠構成剛好四條直線或斜線。

接著為這四條直線的數字，每一條直線佔一列，數字與數字間以英文逗號分隔。

範例：

Sample Input	Sample Output
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25	12
2,4,6,8,10,12,14,16,18,20,22,24,1,3,5,7,9,11,13,15,17,19,21,23,25	2,8,14,20

5	6,12,18,24 4,8,12,6 10,14,18,2 2
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,2 5 2,1,4,3,6,5,8,10,12,14,16,18,20,22,24,7,9,11,13,15,17,19,21,23,2 5	14 1,2,3,4 2,8,14,20 4,8,12,16 10,14,18,2 2
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,2 5 2,1,4,3,6,5,8,7,10,9,12,11,14,13,16,15,18,17,20,19,22,21,24,23,2 5	15 1,2,3,4 6,7,8,9 11,12,13,1 4 1,6,11,16