九十九學年度高級中學資訊學科能力競賽決賽

上機程式設計題

作答注意事項：

一、 對考題有任何疑義，請於考試開始後二個小時之內填寫「問題單」，交付監考人員轉送命題委員提出問題，逾時不予回覆。

二、 第一題和第二題每題 20 分，第三題到第六題各 15 分，共 100 分。

三、 可選擇指定解題語言中任何一種語言解題。

四、 最後繳交編譯後之執行檔限定在 Windows XP 的命令提示字元下執行。

五、 各題執行檔檔名請設定如下：

考生編號_題號.exe

例如：101_1.exe

六、 各題原始碼檔名請設定如下：

考生編號_題號.解題語言附屬檔名

例如：101_1.c

七、 各題輸入資料檔名如下：

in_題號.txt

例如：in_1.txt

八、 各題輸入方式以讀檔方式為之，請以目前工作目錄（current working directory）為讀取路徑。

九、 各題輸出方式為標準輸出（螢幕）。

十、 考生應隨時備份，以防資料流失。隨身碟之備份格式如下：以考生編號加題號為各題的目錄名稱；該目錄下至少須存放該題的執行檔及原始碼檔。例如：在隨身碟 \101_1 這個目錄下，儲存 101_1.c 和 101_1.exe 這兩個檔案。

十一、考試結束後，將不再允許更動及重新編譯程式。

十二、所有發展的程式必須在 2 秒以內於試場內的電腦輸出結果，否則不予計分。

1. 彈珠配置

時間限制：2 秒

問題敘述：

「彈珠排排站」 是園遊會中很受歡迎的腦力激盪遊戲攤位。基本上有 6 個不同的彈珠（彈珠代號為 1, 2, 3, 4, 5, 6）要進行排序。遊戲剛開始時，老闆會先選定一種彈珠排列順序當做解答，爾後遊戲者每次排好彈珠後，老闆會告知有幾個彈珠放對位置，但是不會告知是哪幾個彈珠放對或放錯位置。如果完全正確就可以兌換贈品一份，若猜錯了，則進行下一輪的排序，不過每一次遊戲總共只有七次的排序機會。在任意猜測六次之後，你希望能自動推斷最後一次機會該如何排序來增加獲獎的機率。請寫一個程式，在輸入前六次猜測的紀錄後，計算第七次（最後一次）該採用的彈珠排序方式，也就是找出一組符合前六次猜測結果的彈珠排序順序。

輸入說明：

輸入檔共有 6 行。每行有 7 個以空白隔開的數字，前六個數字代表此輪彈珠排序順序（每個彈珠代號只會出現一次），最後一個數字代表彈珠在正確位置的個數（0～5）。

輸出說明：

請輸出第七輪應該採用的彈珠排序方式。輸入保證至少會有一組答案，若有多組可能答案，請依字典順序（彈珠代號較小的優先）輸出第一組即可。

輸入範例 1：

```
1 2 3 4 5 6 2
4 1 3 2 5 6 3
1 3 4 2 5 6 3
3 4 2 1 5 6 4
2 4 1 3 5 6 2
3 2 4 1 5 6 3
```

輸出範例 1：

```
4 3 2 1 5 6
```

輸入範例 2：

```
6 5 4 3 2 1 0
5 4 3 2 1 6 2
4 2 3 6 5 1 3
1 2 3 4 6 5 4
1 3 2 4 6 5 2
1 2 3 6 4 5 3
```

輸出範例 2：

```
1 2 3 4 5 6
```

2. 圍棋資料庫比對

時間限制: 2 秒

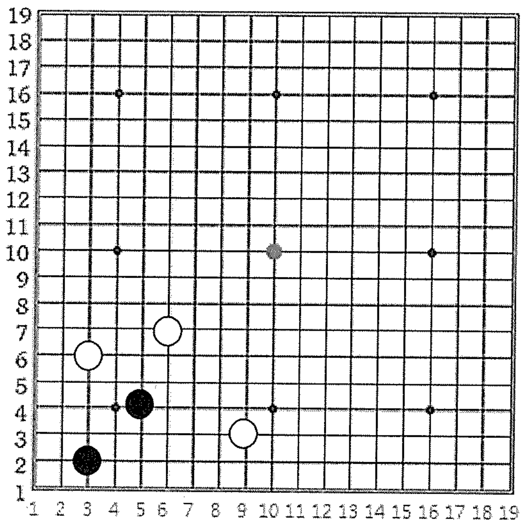
問題敘述：

在電腦圍棋中，為了要能找出最好的對應策略，電腦裡經常儲存了很多著名的棋譜。這些棋譜存放在資料庫中，當玩家和電腦對弈時，電腦會將目前對弈的盤面和資料庫中的棋譜做比對，找出最相似的一個，並且參考找出來的棋譜來決定如何下。

圍棋的棋盤是由 19 條縱線與 19 條橫線交錯所構成，如圖一所示。每一手棋都可以用三個數字 (x, y, c) 來表示，其中 x 和 y 表示這一個棋子的 x 座標和 y 座標，而 c 是表示棋子顏色。我們用數字 0 表示白色，用數字 1 表示黑色。以圖一的例子而言，他有 5 顆棋子，記錄的方法為：

```
3 2 1
9 3 0
6 7 0
5 4 1
3 6 0
```

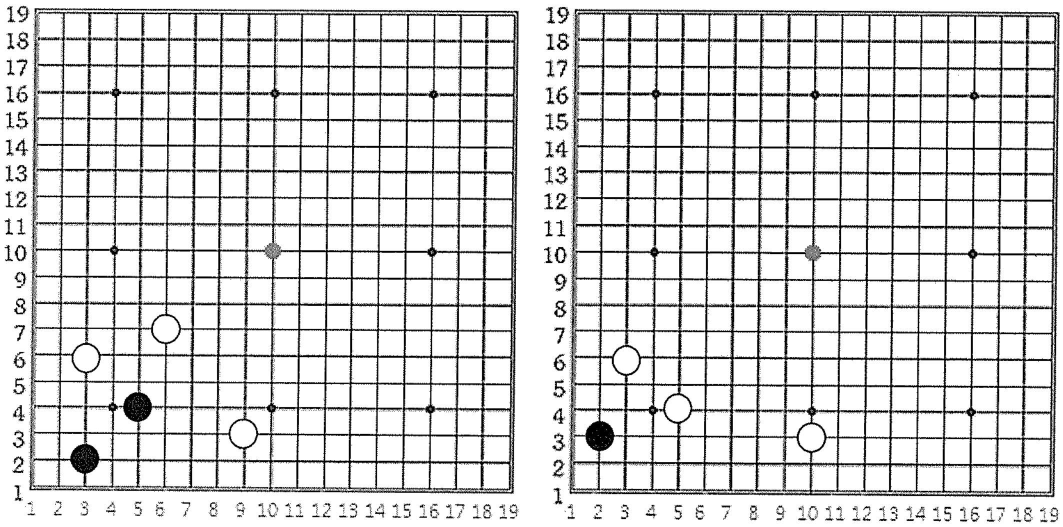
記錄並無固定排列順序，黑子或白子的個數也沒有特別限定(雖然正式棋局中，黑子與白子最多各 180 枚，但是題目中卻不需要去檢查這件事)，對於每個棋譜，同一個位置最多只會出現一次。



圖一

當給定兩個棋盤後，他們的相異程度是這樣計算的。如果一個棋盤在某個座標 (x, y) 有棋子，不論黑子白子，而另外一個棋盤的同一個位置沒有棋子，則相異程度是 +1；如果一個棋盤在某個座標 (x, y) 是黑子，而另外一個棋盤的同一個位置是白子，則相異程度是 +2。其他情況在某個座標 (x, y) 的相異程度是 +0。兩個棋盤的整個相異程度是每個位置相異程度的總和。以圖二的例子來說，

左邊棋盤在位置 (3, 2)，(6, 7) 和 (9, 3) 上有棋子，而右邊棋盤在相同位置上沒有棋子，所以這些位置各自的相異程度都是 +1；反過來說，右邊棋盤在位置 (2, 3) 和 (10, 3) 上有棋子，而左邊棋盤在相同位置上沒有棋子，所以這幾個位置各自的相異程度也是 +1。位置 (3, 6) 上兩個棋盤都是白子，所以相異程度是 +0。而位置 (5, 4) 上，左邊棋盤是黑子，右邊棋盤是白子，所以相異程度是 +2。而其他沒有棋子的地方，相異程度都是 +0。因此，兩盤棋的相異程度是 3+2+2 = 7。



圖二

輸入說明：

第一行為一個整數 n_1，代表第一盤棋譜中棋子的個數， $0 \le n_1 \le 361$。接下來的 n_1 行，每行有三個數字，表示一個棋子的座標和顏色，數字會以一個空白隔開。再下來的一行會是另一個整數 n_2，代表第二盤棋譜中棋子的個數，$0 \le n_2 \le 361$。接下來的 n_2 行，每行有三個數字，表示一個棋子的座標和顏色，數字會以一個空白隔開。

輸出說明：

輸出兩個棋譜的相異程度。

輸入範例 1：

```
2
3 3 1
6 8 0
2
6 8 1
3 2 0
```

輸出範例 1：

```
4
```

輸入範例 2：

```
5
3 2 1
9 3 0
6 7 0
5 4 1
3 6 0
4
10 3 0
2 3 1
3 6 0
5 4 0
```

輸出範例 2：

```
7
```

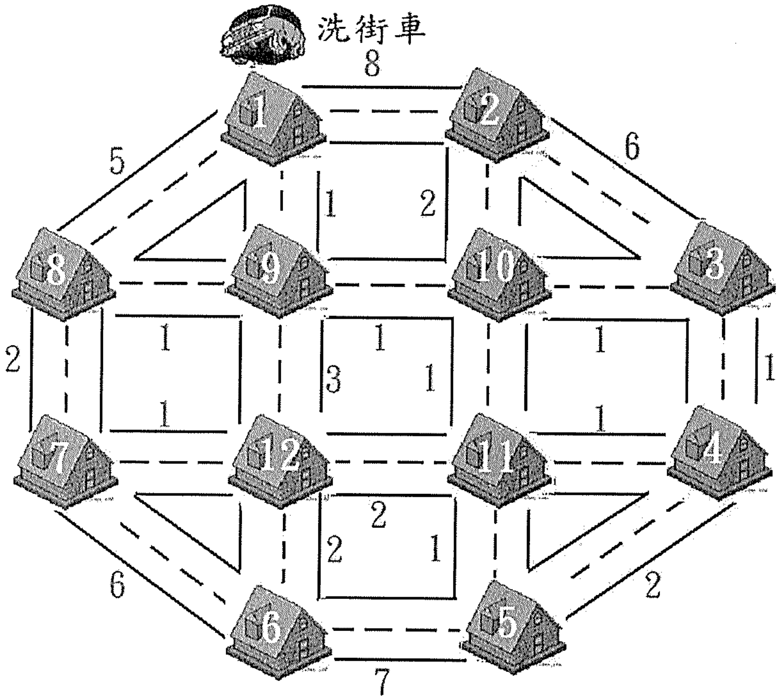
3. 洗街車路線問題

時間限制：2 秒

問題敘述：

市政府派出一輛洗街車欲清洗甲區域的街道，甲區域有 N 個加水站，2 <= N <= 1500。為了方便說明，我們將 N 個加水站名稱以正整數 1, 2, …, N 來表示。N 個加水站有街道來連接，使得洗街車可從任一個加水站出發，經由幾條街道抵達另一個加水站。我們可以用圖形來表示這些加水站跟街道的關係：節點表示加水站；而連接結點的連結線則代表連接兩個加水站之間的街道。我們以符號 (I, J) 來表示連接加水站 I 和加水站 J 的連結線，並稱街道 (I, J) 與加水站 I 和加水站 J ***相接***。每一條連結線 (I, J) 都結合一個權重 w(I, J) 來代表清洗 (I, J) 這條街道所要花費的時間，w(I, J) 為正整數滿足 1 <= w(I, J) <= 888。圖一有 12 個加水站，以數字 1～12 來表示，而街道上的數字則代表清洗此街道所需花費的時間。令 N_{odd} 代表那些與奇數條街道相接的加水站個數，則甲區域有一個重要特性： N_{odd} 為偶數且 0 <= N_{odd} <= 14。在圖一的例子中，N_{odd} = 8，起始加水站為 1。

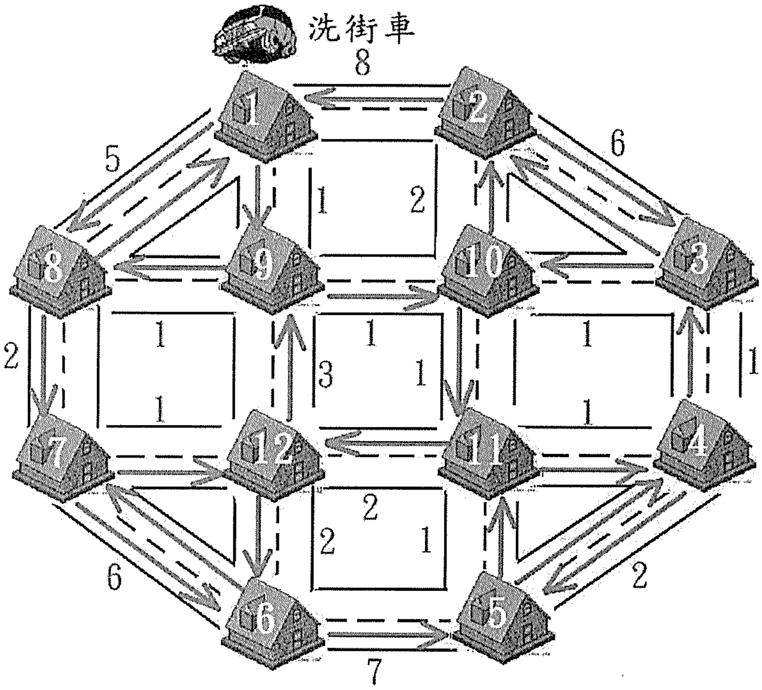
給定一個起始加水站，請寫一個程式計算洗街車從起始加水站出發，把每條街道都清洗過至少一次，再回到原起始加水站所需花費的最短時間為何？注意：這個城市中所有的街道都是雙向道，洗街車洗街時同一個加水站和街道可被洗街車重複經過。



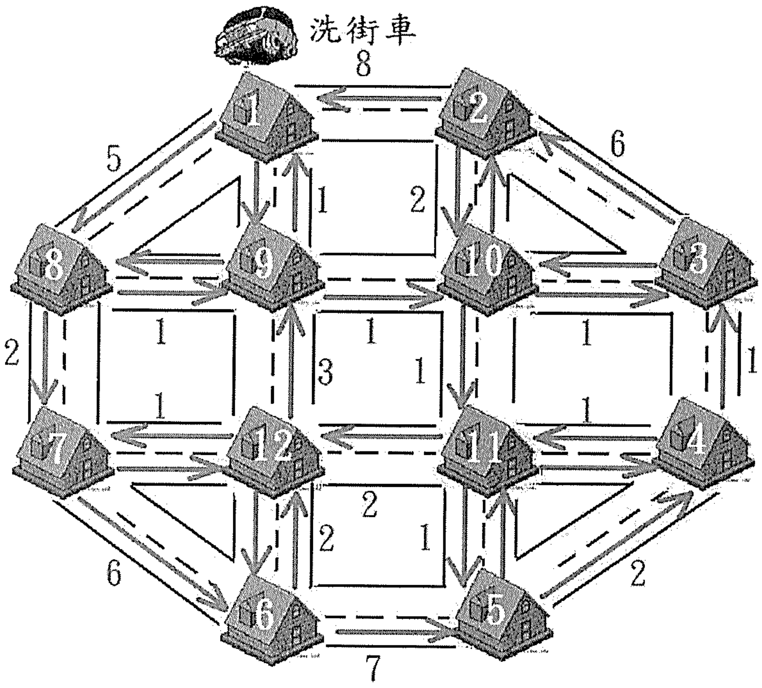
圖一

圖二說明其中一種走法為： 1→8→1→9→8→7→12→6→7→6→5→4→5→11→12→9→10→11→4→3→2→3→10→2→1，可在 73 單位時間將所有街道都清洗

過至少一次，然而此種走法所需的時間並非最短。事實上，此例中花費時間為最短的走法如圖三所示為 1→8→9→1→9→8→7→6→12→7→12→6→5→4→11→5→11→4→3→2→10→3→10→11→12→9→10→2→1，所花費時間為 64 單位。



圖二



圖三

輸入說明：

第一行有三個數字，連續兩個數字之間以空白符號做區格。第一個數字 N (2

<= N <= 1500) 代表圖形的節點個數；第二個數字 M (1 <= M <= N(N-1)/2) 代表圖形的連結線個數(任兩個加水站之間最多只有一條連結線)；第三個數字則代表起始站名稱。從第二行起連續有 M 行，表示 M 條連結線，每行有三個數字，連續兩個數字之間以空白符號做區格：前二個數字代表連結線的兩個端點，第三個數字代表連結線的權重。輸入保證任兩個加水站之間都有路徑相連，N_{odd} 為偶數且 $0 <= N_{odd} <= 14$。

輸出說明：

輸出一個數字代表洗街車所花費的最短時間。

輸入範例 1：

```
12 20 1
1 2 8
1 8 5
2 3 6
1 9 1
2 10 2
8 9 1
9 10 1
10 3 1
8 7 2
9 12 3
10 11 1
3 4 1
7 12 1
12 11 2
11 4 1
7 6 6
12 6 2
11 5 1
4 5 2
6 5 7
```

輸出範例 1：

```
64
```

輸入範例 2：

```
10 9 1
1 2 1
2 3 1
3 4 1
4 5 1
5 6 1
6 7 1
7 8 1
8 9 1
9 10 1
```

輸出範例 2：

```
18
```

輸入範例 3：

```
20 20 1
1 2 1
2 3 1
3 4 1
4 5 1
5 6 1
6 7 1
7 8 1
8 9 1
9 10 1
10 11 1
11 12 1
12 13 1
13 14 1
14 15 1
15 16 1
16 17 1
17 18 1
18 19 1
19 20 1
20 1 1
```

輸出範例 3：

```
20
```

4. 高空煙火

時間限制: 2 秒

問題敘述：

一家專門為大型活動施放煙火的公司，在這次花博標到一件案子，負責沿著河岸施放高空煙火，在河岸每間隔一定距離會設立一個煙火施放點，假設總共有 N 個施放點。煙火的類型主要有兩類，分別為單色系列及彩色系列。單色系列的煙火施放後僅會出現單一顏色的煙火，而彩色系列的煙火施放後會出現多種顏色的彩色煙火。為了讓煙火施放效果生動，兩個施放彩色煙火的施放點之間必須至少有 M 個施放單色煙火的施放點間隔開來，因此施放煙火的方式有許多種可能的施放組合。例如: N=3, M=1 時有 5 種可能施放的方式，即:單單單，彩單單，單單彩，單彩單，彩單彩。

阿博剛好在這家公司打工，給定施放點的個數 N 及 M 的值，阿博的老板要他幫忙算出有多少種可能的施放煙火方式。你的任務是寫一個程式幫阿博計算出所有可能的施放方式。因為答案可能是非常大的數字，為了讓大家能夠簡化運算，所以我們只要求輸出答案最右邊的四位數(以十進位表示)，也就是答案除以 10000 後的餘數(以十進位表示)。

輸入說明：

輸入檔包含兩個正整數 N 及 M，其中 N 代表施放煙火的施放點總數，M 用來代表兩個施放彩色煙火的施放點之間必須至少有 M 個施放單色煙火的施放點間隔開來。其中 1<= M < N <= 3000，並以一或多個空白隔開。

輸出說明：

假設所有可能施放煙火方式的總數為 P，請輸出 P 除以 10000 後的餘數(以十進位表示)。

輸入範例 1：

```
3 1
```

輸出範例 1：

```
5
```

輸入範例 2：

```
19 1
```

輸出範例 2：

```
946
```

輸入範例 3：

```
25 1
```

輸出範例 3：

```
6418
```

5. 貼磁磚

時間限制: 2 秒

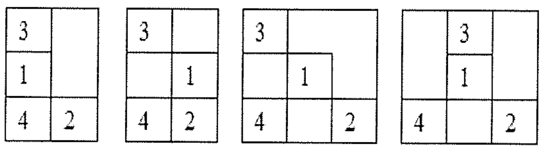
問題敘述：

李先生擁有一棟別墅，由於浴室的地板磁磚已經老舊，因此他想要自己動手重新貼上漂亮的磁磚。他開著車子去買正方形磁磚，當他挑選好想要的磁磚後卻不知道如何擺置才好看，於是老闆就配色上的美觀給了他一些建議，他回到家後，就老闆的建議開始對磁磚擺置做規劃。

假設李先生挑選了 n 個長、寬皆為 w 的正方形磁磚，每個磁磚顏色都不同，並將磁磚編號為 1, 2, …, n。老闆的建議是每個磁磚必須沿著水平線與垂直線擺放(即不能旋轉)，同時任二個磁磚的位置必存在一種相對關係(即上下或左右關係)。假設 $i(1 \leq i \leq n)$、$j(1 \leq j \leq n,\ i \neq j)$為任兩片磁磚的編號，$(x_i,\ y_i)$、$(x_j,\ y_j)$分別表示 i、j 這兩片磁磚擺放後的左下角座標，(x_i+w, y_i+w)、(x_j+w, y_j+w)分別表示它們擺放後的右上角座標。當老闆建議將 i 放在 j 的右邊時，則於完成磁磚擺放後，必須滿足 $x_j+w \leq x_i$；當老闆建議將 i 放在 j 的上面時，則於完成磁磚擺放後，必須滿足 $y_j+w \leq y_i$。此外，相對位置的關係具有可傳遞性，例如：磁磚 i 被建議放在磁磚 j 的上面，磁磚 j 被建議放在磁磚 k 的上面，則磁磚 i 一定被建議放在磁磚 k 的上面；但是不會出現矛盾情形，例如：磁磚 i 被建議放在磁磚 j 的上面，且磁磚 j 被建議放在磁磚 k 的上面，則不會出現將磁磚 k 放在磁磚 i 上面的建議。此外，為了節省浴室空間，李先生希望能將這些磁磚擺置於一個矩形(含正方形)區域內，而且該區域的面積要越小越好。

為了節省時間，李先生希望你能幫忙寫一個程式，規劃出一個符合老闆建議(即每個磁磚必須沿著水平線與垂直線擺放，並滿足 $n(n-1)/2$ 種磁磚相對位置關係)的磁磚擺置方式，而且包圍這個磁磚擺置方式的矩形面積必須是所有滿足老闆建議的擺置方式中的最小值。你的程式必須算出該最小面積，回報給李先生。

以下為一範例：假設李先生購買了 4 個長、寬皆為 1 的正方形磁磚，並將磁磚編號為 1~4，按照老闆的建議，這些磁磚的相對位置關係如下：2 要擺在 4 的右邊、1 要擺在 4 的上面、1 要擺在 2 的上面、3 要擺在 1 的上面、3 要擺在 4 的上面、和 3 要擺在 2 的上面。下圖列出四種滿足老闆建議的擺置方式，其中後二種方式((c)與(d))可分別用面積為 9 的矩形區域將其圍住，而前二種方式((a)與(b))則分別只需用面積為 6 的矩形區域將其圍住。就本範例而言，所有滿足老闆建議的擺置方式中最小的矩形面積為 6，所以你的程式必須回報 6 給李先生。



(a) (b) (c) (d)

輸入說明：

輸入檔的第一行為 n 的值($1 \leq n \leq 500$)，第二行為 w 的值($1 \leq w \leq 10$)，第三行起，連續 $n(n-1)/2$ 行描述磁磚間的相對位置關係，每行有三個以空白分開的整數，分別代表 i、j 和 p 的值，其中前二個整數 i、j 為兩片磁磚編號，第三個整數 p 代表左右或上下關係，當 i 被建議放在 j 的右邊時，p 為 0，而當 i 被建議放在 j 的上面時，則 p 為 1。

輸出說明：

用一行輸出回報給李先生的面積值。

輸入範例 1：

```
4
1
2 4 0
1 4 1
1 2 1
3 1 1
3 4 1
3 2 1
```

輸出範例 1：

```
6
```

輸入範例 2：

```
6
3
2 1 0
3 2 0
3 1 0
6 4 0
4 1 1
5 1 1
4 2 1
5 2 1
4 3 1
```

```
5 3 1
5 4 1
6 1 1
6 2 1
6 3 1
5 6 1
```

輸出範例 2：

```
81
```

6. 雨量趨勢

時間限制: 2 秒

問題敘述：

台灣近年來經常豪雨成災。每遇有豪雨，新聞記者就將以往降雨紀錄拿出，並以聳動字眼云之，如「30 年來最大降雨量」、「40 年來首見」等。李先生是一位科學家，認為用直覺讀取數據容易造成偏頗，他想用更科學的方式來解讀降雨量，希望能找出降雨量長期趨勢。在觀察一連串的降雨量數據後，李先生想找出最長的遞增序列，以代表長期趨勢。在他所找到的長期趨勢資料中，一般而言是逐步上升，但某次破紀錄的最大雨量之後，若有稍大雨量(雖未破紀錄)，亦視為遞增序列之內容。換言之，他欲找出最長近乎遞增序列。

舉例來說，李先生的雨量數據有 12 個，如下：`

9, 15, 3, 14, 14, 8, 10, 11, 17, 15, 14, 13

另外設定一個 E 值，代表稍大雨量與長期趨勢目前最大雨量可以允許的差值。假設 E=1，則可以找出下列五組最長近乎遞增序列，其長度均為 6：

(9, 15, 14, 14, 15, 14), (9, 8, 10, 11, 15, 14), (3, 8, 10, 11, 15, 14),

(9, 8, 10, 11, 14, 13), (3, 8, 10, 11, 14, 13)

又若 E=0，則最長近乎遞增序列有四組(3, 8, 10, 11, 13), (3, 8, 10, 11, 14), (3, 8, 10, 11, 15), (3, 8, 10, 11, 17)，其長度均為 5。

再看另一個例子，假設雨量數據為 3, 4, 4, 6, 5, 4。若 E=0，則有三組最長近乎遞增序列：(3, 4, 4, 6), (3, 4, 4, 5), (3, 4, 4, 4)，長度均為 4。若 E=1，則有二組：(3, 4, 4, 6, 5), (3, 4, 4, 5, 4)。若 E=2，則有一組：(3, 4, 4, 6, 5, 4)。本題的答案只要輸出長度即可。

輸入說明：

輸入檔分成三部分，首先第一行有兩個正整數 n 與 m，其中 1<= n <= 3500，1<= m <= 3；第二行有 n 個雨量數據；第三行有 m 個 E 值。雨量與 E 值均為非負整數，最大值不超過 999999。每一行的數字與數字間會以空白隔開。

輸出說明：

分別輸出 m 個 E 值所對應的最長近乎遞增序列之長度。這 m 個整數請輸出於一列，且相鄰兩個整數之間以一個空白隔開。

輸入範例 1：

```
12 3
9 15 3 14 14 8 10 11 17 15 14 13
0 1 999990
```

輸出範例 1：

```
5 6 12
```

輸入範例 2：

```
4 3
1 0 999900 999800
1000 0 1
```

輸出範例 2：

```
4 2 3
```