

## 高中組（參考範例）

### 第一題：小數加法

題目內容：溫教授最近在上計算機概論，要教學生使用十進位制以外的不同進位制度，例如用位元就能表示的二進位、方便工程師閱讀的十六進位等等。不過，最近的學生沒有狼性，他怎麼教學生都聽不懂，溫教授為此非常煩惱，於是走進附近的小 7 超商想買杯咖啡。

看著架上商品的價錢之際，教授忽然靈機一動，言教不如身教，學生如果沒有從做中學，怎麼有辦法擺脫習慣的十進位使用其他的進位制呢？於是教授決定偷偷委託親朋好友在校內開設一間新的「真·小 7 超商」，裡面所有的價格都用七進制來表示，其中有些商品比一般的小 7 超商還貴，有些則便宜很多，這樣子學生為了撿便宜就會嘗試來店內挖寶。

不過，超商開幕後，溫教授卻發現連他自己都購物得很吃力。身為教授連自己都算錯的話就太糗了，只好委託你幫他開發一個能夠幫他做七進制加法的程式。因為商品的價格當然都是正的，所以價錢只會是正的整數或小數。順帶一提，在小 7 超商內結帳所的貨幣當然也是採用七進制的，因此可以不必擔心如何付款的問題。

輸入說明：輸入共有兩列，每一列包含一個七進位制下的數值。輸入中每一個七進位制下的數值可能為七進制正整數或是大於零的七進制小數，格式如下：若是七進制正整數，則以由阿拉伯數字 0, 1, 2, 3, 4, 5, 6 所組成的連續字串表示，且不會以 0 為開頭；若是大於零的七進制小數，則以由阿拉伯數字 0, 1, 2, 3, 4, 5, 6 與恰好一個小數點所組成的連續字串表示，且除了以「0.」開頭的情況下允許以 0 為開頭外，不會以 0 或小數點為開頭或結尾。

輸出說明：請輸出一列，其中包含一個七進位制下的數值，表示在七進位制運算下，輸入中兩個數的總和，也須以七進制表示，若此總和可以用七進制正整數表示，則請將該數值表示為以阿拉伯數字 0, 1, 2, 3, 4, 5, 6 所組成的連續字串，且不得以 0 開頭；否則，請將該數值以七進制小數表示，也即表示為以阿拉伯數字 0, 1, 2, 3, 4, 5, 6 與恰好一個小數點所組成的連續字串，且除了以「0.」開頭的情況下允許以 0 為開頭外，不得以 0 或小數點為開頭或結尾。

範例輸入一：123 45

範例輸出一： 201

範例輸入二： 3 4.2

範例輸出二： 10.2

範例輸入三： 10.6 0.1

範例輸出三： 11

評分說明：正式評分所使用的測試資料共分為 7 組，每組測試資料佔 2 分，滿分 7\*2 分。對於其中至少 3 組測試資料，輸入中的數值皆不含小數點。對於所有測試資料，輸入中的數值長度均不超過 7 位(小數則包含小數點)。

## 第二題：最短路徑

題目內容：

達狗是一座長方形的幸福城市，道路四通八達，而且都與達狗的邊界平行或垂直，任意兩條相鄰且平行的道路，或者邊界與最近的平行道路的間距都恰為 1 單位距離。因為這樣的特性，我們用二維平面來描述達狗的任意路口：坐標平面上  $x$  軸向東側為正、 $y$  軸向北側為正，並將達狗西南側端點之坐標位置定為  $(x_1, y_1)$ ，東北側端點之坐標位置定為  $(x_2, y_2)$ ，也就是說在坐標點  $(x, y)$  時，向東側走 1 單位距離，會到達  $(x + 1, y)$ ；向北側走 1 單位距離，則會到達  $(x, y + 1)$ 。

達狗的交通非常便利：每當走到四周的邊界，就會被傳送至與該邊界平行之另一側邊界之對應位置。舉例來說，若走到北側邊界  $(x, y_2)$  上，會被傳送到  $(x, y_1)$  的位置；若走到南側邊界  $(x, y_1)$  上，會被傳送到  $(x, y_2)$  的位置。類似地，若走到東側邊界  $(x_2, y)$  上，會被傳送到  $(x_1, y)$  的位置；走到西側邊界  $(x_1, y)$  上，會被傳送到  $(x_2, y)$  的位置。

現有兩個人身在達狗的不同路口，想要約在一個不在邊界上的路口會合，並且希望兩人所行走的距離總和愈短最好。兩人經過一番思考，驚覺這正是傳說中的最短路徑問題！於是已經學過最短路徑演算法的你，自告奮勇想幫忙他們計算這個問題。

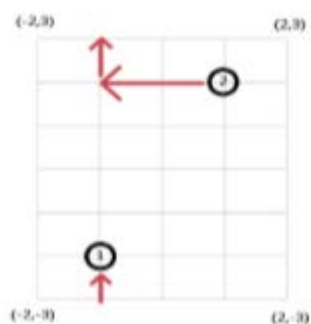
輸入說明：

輸入的第一列為兩個整數  $x_1, y_1$ ，表示達狗西南側端點之坐標位置為  $(x_1, y_1)$ 。

輸入的第二列為兩個整數 $x_2, y_2$ ，表示達狗東北側端點之坐標位置為 $(x_2, y_2)$ 。  
 輸入的第三列為兩個整數 $x_3, y_3$ ，表示第一個人所站的坐標位置為 $(x_3, y_3)$ 。  
 輸入的第四列為兩個整數 $x_4, y_4$ ，表示第二個人所站的坐標位置為 $(x_4, y_4)$ 。  
 輸入的兩人位置不同，且 $x_1 < x_3 < x_2$ ； $x_1 < x_4 < x_2$ ； $y_1 < y_3 < y_2$ ； $y_1 < y_4 < y_2$ 。

輸出說明：

請輸出一列，其中包含一個正整數，表示這兩個人所行走的最小距離總和。



範例輸入一：

0 0

10 10

5 5

6 7

範例輸出一：

3

範例說明一：第一個人由 $(5, 5)$ 開始往北走 2 單位，第二個人由 $(6, 7)$ 往西走 1 單位，即可到達相同的位置 $(5, 7)$ ，行走距離的總和是 3 單位，此為距離總和最短之走法。

範例輸入二：

-2 -3

2 3

-1 -2

1 2

範例輸出二：

4

範例說明二：

如圖北方朝上，第一個人維持不動，第二個人由 $(1, 2)$ 往西走 2 單位，再向北走 2 單位，途中當行走 1 單位時，會先到達 $(-1, 3)$ 之北側邊界，傳送至 $(-1, -3)$ 後，再行走一單位，即可到達 $(-1, -2)$ ，此為與第一個人相同的位置，行走距離的總和是 4 單位，此為距離總和最短之走法。

評分說明：

正式評分所使用的測試資料共分為 10 組，每組測試資料佔 1 分，滿分 10 分。對於至少 4 組測試資料， $x_1 = y_1 = 0$ ，輸入中的數值皆為非負整數，且值皆小於 1000。對於所有測試資料，輸入中的所有數值之絕對值皆小於  $2^{31}$ 。

第三題：遞迴搜索

題目內容：

溫教授正在用心的出著程式設計競賽的考題。為了讓學生能更快地想到解法，避免給予學生太大的挫折，在幾經思考後，溫教授決定在題目名稱中放上與該題相關的解題方法，但不幸地遇到了一些小麻煩。

有些題目可能會與數種不同的解題方法有相關，而也可能有些不同的題目不巧的都與某種解題方法所相關。溫教授希望對每一題，都挑選恰好一種與該題相關的解題方法，來訂為該題目的名稱。此外，為了避免混淆，溫教授希望不同題目的名稱不可以重複。

教授已經知道每一題與哪些解題方法有相關，若想要在滿足上述條件限制的情形下，對每一題按照上述規則訂定其名稱，是否能找出至少一種符合規則的解法呢？

為了降低決定題目名稱的複雜度，每一套題目都各自獨立，不需要考慮不同套題目之間的題目名稱是否有重複。對每一套題目而言，只需要訂定恰好三題的題目名稱，且每一題恰好都與兩種解題方法有相關。

教授苦思良久後，決定請已經征服遞迴函數與回溯法搜索的你，協助他訂定每一套題目中三題的題目名稱。

輸入說明：

輸入的第一列為正整數  $T$ ，表示總共有多少套題目需要決定題目名稱。接著共有  $T$  列，每一列各自表示一套題目，總共有三題，也即每一列總共包含 6 個正整數，相同的數表示著相同的解題方法，反之亦然。其中每一列的第 1, 2 個數所表示的解題方法與第一題有相關；第 3, 4 個數所表示的解題方法與第二題有相關；第 5, 6 個數所表示的解題方法與第三題有相關。

輸入滿足  $T \leq 1000$ ，且所有表示解題方法的正整數皆不超過 100。

輸出說明：

對於每一套題目，請依序輸出一列，若對於該套題目，可以依照題意不重複的訂定每一題的題目名稱，請輸出 Yes，否則輸出 No。

範例輸入一：

1

1 2 3 4 5 6

範例輸出一：

Yes

範例說明一：

三題的題目名稱可以分別訂為 1、4、5 符合條件且不重複。

範例輸入二：

2

1 2 1 2 1 2

2 1 2 1 2 1

範例輸出二：

No

No

範例說明二：

均不存在不重複之題目名稱訂法。

範例輸入三：

1

1 2 1 3 2 3

範例輸出三：

Yes

範例說明三：

三題的題目名稱可以分別訂為 2、1、3 符合條件且不重複。

評分說明：正式評分所使用的測試資料僅有一組，滿分 10 分。

第四題：二維陣列

題目內容：

有一座可以視為二維的方格平面的長方形城市，由上而下共有 $N$ 列，由左至右共有 $M$ 行，總共劃分為 $N \times M$ 個方格。這個城市因為治安很差，所以每個格子中都有機會出現野生的甩棍可以撿拾以方便鬥毆。你想要撿一些甩棍，但你知道城市的其中一些方格設有警察局，如果警察局內的警察在這個方格或與其同一行或同一列的任何方格偵測到有人在撿拾甩棍，都會毫不留情地把撿拾的人預防性羈押，所以為了安全，你絕對不會去那些方格撿拾甩棍。你想要知道在上述條件下，你還可以去那些方格撿拾甩棍。

輸入說明：

輸入的第一列依序為正整數 $N, M$ ，表示這座城市由上而下共有 $N$ 列，由左至右共有 $M$ 行，總共劃分為 $N \times M$ 個方格。

接著共有 $N$ 列，每列為一個長度恰為 $M$ 的字串，每個字元則可能為「.」(ASCII 編碼 46) 或「#」(ASCII 編碼 35)，若其中第 $i$ 列的第 $j$ 個字元為「#」，表示城市中第 $i$ 列第 $j$ 行的方格有裝設甩棍毀滅器，否則表示沒有。除此之外，該列不會有其他的字元。

輸出說明：

請輸出 $N$ 列，每列為長度恰為 $M$ 的字串，對於第 $i$ 列的第 $j$ 個字元，若城市中第 $i$ 列第 $j$ 行的方格有機會可以撿到甩棍，則請輸出「X」，否則請輸出「#」。請勿輸出其他的字元或空白。

範例輸入一：

```
2 2
#. .
```

範例輸出一：

```
##
#X
```

範例輸入二：

3 2

##

..

#.

範例輸出二：

##

##

##

範例輸入三：

1 4

..#.

範例輸出三：

####

評分說明：

正式評分所使用的測試資料共分為 10 組，其輸入條件限制及配分列舉如下，每組測試 資料完全正確得該組測試資料配分，否則不給分，滿分 10 分。

1.  $N \leq 10$   $M \leq 10$
2.  $N \leq 50$   $M \leq 50$
3.  $N \leq 300$   $M \leq 300$  至多 300 個#
4.  $N \leq 300$   $M \leq 300$
5.  $N \leq 300$   $M \leq 300$
6.  $N \leq 3000$   $M \leq 3000$  至多 3000 個#
7.  $N \leq 3000$   $M \leq 3000$  至多 30000 個#
8.  $N \leq 3000$   $M \leq 3000$  至多 300000 個#
9.  $N \leq 3000$   $M \leq 3000$
10.  $N \leq 3000$   $M \leq 3000$