

第一題：四點共線 (collinearity)

執行時間: 1 秒

問題描述

給定平面上相異 n 個點，編號由 1 至 n 。針對任意四點，定義該四點的「組號」為一四維坐標，由該四點的編號由小而大排序而成；例如編號為 1, 3, 4, 10 的四點之組號為 (1, 3, 4, 10)；編號為 22, 101, 11, 49 的四點之組號為 (11, 22, 49, 101)。兩組號的大小比較，則依字典序判斷；例如 (1, 5, 10, 12) 較 (2, 3, 4, 6) 來的小。請判斷輸入的 n 點中是否有某四個點共線，若沒有，請輸出 0；若有，請輸出共線的四點中最小的組號。

輸入格式

第一行為正整數 n 。接下來 n 行，依序表示點 1 至點 n ；每行有兩個整數表示該點之 x 與 y 坐標，此二整數介於 $-(10^4)$ 和 10^4 間。此 n 點皆為不同點，同一行的兩數字間以一空白區隔。

輸出格式

若無四點共線，請輸出 0；若有四點共線，則輸出共線的四點中組號最小的四個數字，兩數字間以一空白區隔。

輸入範例一 6 33 33 11 17 2 2 4 4 5 5 -1 -1	輸出範例一 1 3 4 5
輸入範例二 4 -1 0 2 0 3 0 2 1	輸出範例二 0

評分說明

本題共有三組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	20	$n = 4$ 。
2	25	$n \leq 100$ 。
3	55	$1000 \leq n \leq 3000$ 。

第二題：韓信點兵 (soldier)

執行時間: 1 秒

問題描述

韓信點兵，每 a_1 個一數剩 b_1 個人，每 a_2 個一數剩 b_2 個人， \dots ，每 a_n 個一數剩 b_n 個人；已知韓信確實有兵 (人數不為零)，請問韓信「最少」有多少兵？另外在點兵的時候，可能會點錯，所以可能無解，如果無解的時候，請回答 -1。給定 n 與 a_1, \dots, a_n 皆為正整數；對任意介於 1 到 n 的整數 i 而言， b_i 為整數，且滿足 $0 \leq b_i < a_i$ 。已知答案小於 2^{64} ，且 a_1, a_2, \dots, a_n 的最小公倍數小於 2^{63} 。

例如：每 3 個一數剩 1 個人，每 5 個一數剩 3 個人，則考慮大於等於 1 的解，其最小解為 13，所以韓信最少有 13 個兵。

輸入格式

第一行為正整數 n ，接下來 n 行，每行兩個整數 a_i 與 b_i 。

輸出格式

輸出一整數，代表韓信最少有多少兵。如果點兵時出錯造成無解，請輸出 -1。

輸入範例一 2 3 1 5 3	輸出範例一 13
輸入範例二 3 7 1 8 2 9 3	輸出範例二 498

評分說明

本題共有五組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	18	$n = 2$ ，點兵沒出錯， a_1, a_2, \dots, a_n 皆不超過 256。
2	19	$n = 5$ ，點兵沒出錯， a_1, a_2, \dots, a_n 皆不超過 1000。
3	20	$n = 2$ ，點兵沒出錯。
4	21	$n = 3$ ，點兵沒出錯。
5	22	$n = 5$ ，點兵可能出錯。

第三題：小軒小羽踩水坑 (puddle)

執行時間: 1 秒

問題描述

小軒跟小羽是兩位好朋友，小軒家有個矩形的院子鋪滿了正方形的磁磚，下過大雨後院子會積水，小軒想要玩踩水坑的遊戲，於是把某些區域圍成了大大小小的水坑，但是小羽喜歡把小軒用來圍水坑的阻擋物拆掉。本題要請你寫一支程式計算有多少水坑以及最大水坑的面積。

小軒在圍水坑時，某些磁磚的位置整個都是阻擋物，而另外那些磁磚的位置則完全是有水，而每一個磁磚的面積都是 1。院子的積水情形可以用一個矩陣來表示，就類似下面那張方格圖所示。矩陣的每一個橫排是一列，直排是一行，矩陣左方是列的編號，而上方是行的編號，第 r 列第 c 行的格子以 $A[r, c]$ 表示。矩陣內標示為 0 格子裡都是水，標示為 1 的格子是阻擋物完全沒有水。有水的格子跟它的上、下、左、右四個方向的有水格子就被看成是連通的，任何兩個有水的格子直接或間接相通就屬於同一個水坑。請注意，斜角方向不算連通。

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

以上面的範例來說，此院子被分成 6 個水坑，每個水坑包含的格子如下，其中最大的水坑是第 5 個水坑，面積是 5，因為它佔了 5 個格子：

1. $A[1, 1]$
2. $A[1, 3]$
3. $A[1, 6], A[1, 7], A[2, 6], A[3, 6]$
4. $A[3, 1], A[3, 2]$
5. $A[4, 3], A[4, 4], A[4, 5], A[5, 5], A[5, 6]$
6. $A[5, 1], A[5, 2]$

在小軒把水坑圍好後，他的好朋友小羽就開始動作，小羽每次把其中一個磁磚的阻擋物拆掉，並且讓該位置變成有水，也就相當於把某個矩陣位置設為成 0。小羽的動作可能會製造一個新的水坑，也可能把原來不相通的水坑變成相通。假設小羽開始動作前的水坑數量為 P_0 ，最大水坑面積是 Q_0 。小羽做完第 1 個動作後的水坑數與最大面積變成 P_1 與 Q_1 ，做完前兩個動作後則變成 P_2 與 Q_2 ，…。小羽總共做了 k 次動作，你的程式要計算出 $\sum_{i=0}^k P_i$ 以及 $\sum_{i=0}^k Q_i$ 。

輸入格式

輸入的第一行有三個整數，依序是 M 、 N 、 k ，代表矩陣是 M 列 N 行，以及小羽的動作次數， $M \leq 200$ 、 $N \leq 500$ 、 $k \leq 50,000$ 。第二行開始有 M 行輸入資料，每行有 N 個 0 或 1 的數字，代表矩陣的內容，順序是由上而下，由左至右(可參考下面範例二中的上述例子)。矩陣資料之後有 k 行，依序代表小羽每一次動作的位置，每行兩個整數 r 與 c 表示動作的位置是將 $A[r, c]$ 設成 0，此位置原本必定是 1。同一行數字之間都是以空白間隔。

輸出格式

輸出兩行，依序是 $\sum_{i=0}^k P_i$ 以及 $\sum_{i=0}^k Q_i$ 。注意，答案可能超過 2^{32} ，但不會超過 2^{60} 。

輸入範例一 1 5 0 0 1 1 1 0	輸出範例一 2 1
輸入範例二 5 7 2 0 1 0 1 1 0 0 1 1 1 1 1 0 1 0 0 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 2 4 4 6	輸出範例二 19 20

範例一說明：此範例是下面子題 1 限制內的範例，矩陣只有一列，動作數為 0。有兩個水坑，面積都是 1，所以最大面積也是 1。

範例二說明：此矩陣就是題目中所舉的例子，如題目中說明，在小羽動作前有 6 個水坑，最大面積是 5。第一個動作的位置是 $A[2, 4]$ ，此動作增加了一個面積為 1 的水坑，所以此時水坑數是 7，最大面積還是 5。第二個動作位置是 $A[4, 6]$ ， $A[4, 6]$ 改成 0 後，會把原來的兩個水坑連通，因此水坑數量變成 6，最大面積變成 10。因此，水坑數量加總是 $6+7+6=19$ ，最大面積總和是 $5+5+10=20$ 。

評分說明

本題共有三組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	20	$M = 1$ 且 $k = 0$ 。
2	40	$k = 0$ 。
3	40	無額外限制。

第四題：雲霄飛車 (rollercoaster)

執行時間: 1 秒

問題敘述

最近 X 大學開了一間附屬遊樂園，搭建了一個雲霄飛車；然而營運了一段時間後，園方覺得目前的軌道設計不夠刺激，想調整雲霄飛車的軌道使其變得更「刺激」。雲霄飛車的軌道是由一根根的柱子所撐起；園方對「刺激」的定義為「依高度最高的某根柱子將軌道區分為前後兩半，軌道前半部柱子的高度必須為一遞增數列（允許相同高度的柱子），後半部的柱子高度則為一遞減數列（同樣允許相同高度的柱子）」。

至於「前半部」和「後半部」該有幾根柱子則無特別限制。例如，(1, 20, 30, 2, 1), (1, 1, 1, 3, 5, 4, 2), (9, 4, 3, 2, 1), (3, 7, 8) 都符合園方對雲霄飛車「刺激」的要求。

精確來說，若有一由 n 根柱子，高度分別為 h_1, h_2, \dots, h_n ，搭建成的「刺激」雲霄飛車，若且唯若存在一個 1 以上， n 以下的整數 k ，滿足

- 對於 1 以上不滿 k 的任意整數 i ， $h_i \leq h_{i+1}$ 且
- 對於 k 以上不滿 n 的任意整數 i ， $h_i \geq h_{i+1}$ 。

然而，目前園方的人手不足，剩餘的人力只能透過把柱子與相鄰的柱子交換來完成雲霄飛車的軌道調整。請你幫忙園方算出，最少須幾次交換才能調整出「刺激」的雲霄飛車。例如，原本的柱子的高度排列為 (7, 5, 6, 3)，藉由一次交換，交換 5 與 6 或交換 7 與 5，可完成「刺激」的雲霄飛車。

輸入格式

輸入有兩行，第一行有一個整數，表示 n ， $1 \leq n \leq 10^5$ 。第二行有 n 個整數 a_1, a_2, \dots, a_n ，代表調整前柱子由前而後依序的高度；任意的高度 a_i 皆滿足 $1 \leq a_i \leq 10^5$ 。

輸出格式

請輸出園方最少需要的相鄰柱子交換次數，使得雲霄飛車的軌道能變為「刺激」。

輸入範例一 3 7 5 6	輸出範例一 1
輸入範例二 6 8 7 2 5 4 6	輸出範例二 4
輸入範例三 7 3 1 4 1 7 20 2	輸出範例三 3

評分說明

本題共有四組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	11	$1 \leq n \leq 3$ ，且柱子高度皆不相同。
2	22	$1 \leq n \leq 10$ ，柱子高度有可能相同。
3	33	$1 \leq n \leq 100$ ，柱子高度有可能相同。
4	34	無額外限制。

第五題：傳真修復 (fixing)

執行時間: 2 秒

問題描述

早期人們常用傳真機來傳遞文件，將一文件掃描成圖片(由許多像素構成)，每個像素只有兩個可能的數值：0(代表黑色)或 1(代表白色)。掃描後的文件常帶有雜訊，原本某些黑色的像素在掃描後卻變成白的，或反之。舉例來說，圖一顯示一個帶有雜訊的掃描文件，從中我們隱約看到了大寫字母 A。

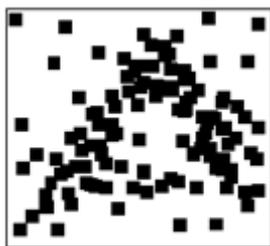
這些雜訊可以透過軟體去除，進而修復傳真文件，這過程是一個最佳化問題。在修復後的文件中，每個像素仍非黑即白，修復的程序要考量兩個面向：

1. 我們希望盡量保持原本文件的內容，也就是每個像素能盡量維持原本的數值(黑或白)
2. 我們希望能「淨化」文件，也就是在修復結果中鄰近的像素有相同的值。

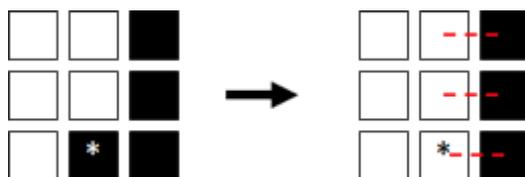
因此，我們定義了兩種修復成本如下：

1. 若某像素的數值因修復而改變(黑變成白或是白變成黑)，這個像素的修復成本為 2。舉例來說，圖二的每一個格子代表一個像素，將圖二(左)修復為圖二(右)的第一種成本為 2，因為其中一個像素的值改變了(以星號標註)。
2. 第二種成本計算在修復後文件中的不連續性，與原始文件無關。在修復後的文件中，若任兩個鄰近的像素(考量上、下、左、右共四個位置)有不同的值(一黑一白)，這個組合的成本為 1。以圖二為例，第二種修復成本為 3，因為在水平及垂直方向共計 3 組像素變換(以虛線標註)。

一個文件的修復成本為以上兩種成本的總和，而一個好的修復方法應該有低的修復成本。請寫一個程式，輸入一個帶雜訊的掃描文件，計算成本最低的修復方式。因為可能存在多種最低成本的修復方式，因此，請輸出最低成本即可。注意：圖二為解釋成本的範例，不一定是最佳解。



圖一



圖二

輸入格式

1. 輸入的第一行有兩個正整數 H ($1 \leq H \leq 30$) 與 W ($1 \leq W \leq 30$)，代表輸入文件掃描成 $H \times W$ 個像素。
2. 接下來有 H 行，每行有 W 個像素值 (0 或 1)。

輸出格式

輸出為一整數，代表最低成本。

輸入範例一 3 3 1 1 0 1 1 0 1 0 0	輸出範例一 4
輸入範例二 1 5 0 0 1 0 1	輸出範例二 3

評分說明

本題共有三組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	23	$H \times W < 20$ 。
2	38	$H = 1$ 。
3	39	無額外限制。