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	輸出範例一
6 33 33 11 17 2 2 4 4 5 5 -1 -1	1 3 4 5
輸入範例二	**輸出範例二**
4 -1 0 2 0 3 0 2 1	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	$n \le 100$。
3	55	$1000 \le n \le 3000$。

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，⋯，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 $a_1, \ldots, a_n$ 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 $0 \le b_i < a_i$。已知答案小於 2^{64}，且 $a_1, a_2, \ldots, a_n$ 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	輸出範例一
2 3 1 5 3	13
輸入範例二	**輸出範例二**
3 7 1 8 2 9 3	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，$a_1, a_2, \ldots, a_n$ 皆不超過 256。
2	19	$n = 5$，點兵沒出錯，$a_1, a_2, \ldots, a_n$ 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，$M \le 200$、$N \le 500$、$k \le 50{,}000$。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	輸出範例一
1 5 0 0 1 1 1 0	2 1
輸入範例二	**輸出範例二**
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	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的更「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 $h_1, h_2, ..., h_n$，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，$h_i \le h_{i+1}$ 且
- 對於 k 以上不滿 n 的任意整數 i，$h_i \ge h_{i+1}$。

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，$1 \le n \le 10^5$。第二行有 n 個整數 $a_1, a_2, ..., a_n$，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 $1 \le a_i \le 10^5$。

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	輸出範例一
3 7 5 6	1
輸入範例二	**輸出範例二**
6 8 7 2 5 4 6	4
輸入範例三	**輸出範例三**
7 3 1 4 1 7 20 2	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	$1 \le n \le 3$，且柱子高度皆不相同。
2	22	$1 \le n \le 10$，柱子高度有可能相同。
3	33	$1 \le n \le 100$，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：

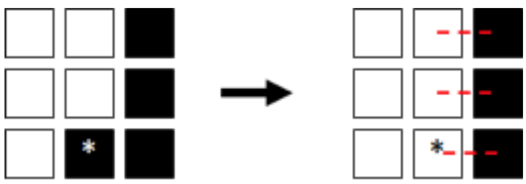
1. 我們希望盡量保持原本文件的內容，也就是每個像素能盡量維持原本的數值(黑或白)
2. 我們希望能「淨化」文件，也就是在修復結果中鄰近的像素有相同的值。

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 $(1 \le H \le 30)$ 與 W $(1 \le W \le 30)$，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	輸出範例一
3 3 1 1 0 1 1 0 1 0 0	4
輸入範例二	**輸出範例二**
1 5 0 0 1 0 1	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。

第一題：浪人 (Walking)

問題敘述

去年九月初的時候，有將近四百位狼人在日本的茨城縣出沒，不斷在狼人的兩個型態——狼形與人形之間相互轉換。身處於台北市的你，覺得在兩個型態之間轉換並不是一件很新潮的事情。

「如果能在平行宇宙當中行走，那該有多好！」坐在電腦前看著《愛麗絲漫遊量子奇境》小說發呆的你，覺得如果能同時存在於兩個世界中隨意行走，似乎是一件能夠展現出現代人浪費才能做出來的事情。身為一個浪人，你決心要在現實世界和平行世界中的台北市漫遊。

現實世界中的台北市有 N 個里，還有 N-1 條單向道路，每一條單向道路都是從某個里直達另一個里。為了方便起見，我們把這 N 個里編號為 $1, 2, \ldots, N$。根據你縝密地觀察，發現現實世界中的任何一個里，都可以從編號為 1 的里經過若干條單向道路到達。

「這太不可思議了。我決定要設計一個平行世界裡的台北市，打破這個傳統！」於是你設計出了一個平行世界中的台北市。這個台北市也有 N 個里、還有 N-1 條單向道路。在這個平行世界中的台北市，雖然無法保證從編號為 1 的里總可以沿著某些單向道路到達其他里，但是，對於任何一個里，總存在一系列的單向道路，如果允許逆向行駛的話就可以到達。

身為一個浪費才能的人，你打算隨機從兩個世界中的同一個里出發，然後在兩個平行世界中隨意沿著單向道路走動——不一定要每次都一起走到另一個里，可以分開行走，甚至可以只有其中一個平行世界的人移動就好——期待著兩個世界中的彼此又再度匯聚於同一個里。這種事情真的常常發生嗎？

於是，思緒回到電腦前的你，決定在這個充滿戰鬥力的一天，寫一個程式，計算看看有多少個里的配對 (S, E)，其中 S 與 E 不同，並使得從兩個平行世界的 S 里出發，隨意走動之後，最終都能走到 E 里。

輸入格式

每筆測試資料的第一列有一個數字 N ($1 \leq N \leq 200000$)。接下來有 $2N$-2 列，每一列有兩個數字 x, y ($1 \leq x, y \leq N$，$x \neq y$)，首 N-1 列代表著現實世界中從 x 里通往 y 里的道路，末 N-1 列則代表平行世界中的道路。你可以假設輸入保證滿足題目敘述的要求。

輸出格式

對於每一筆測試資料，輸出題目要求的 (S, E) 配對的數量。

輸入範例 1	**輸出範例 1**
3 1 2 2 3 1 3 3 2	2

輸入範例 2	**輸出範例 2**
3 1 2 1 3 3 2 2 1	0

輸入範例 3	**輸出範例 3**
5 1 2 2 3 3 4 4 5 2 1 3 1 3 4 5 4	1

評分說明

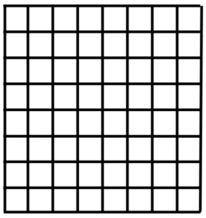
本題共有 4 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	8	$1 \leq N \leq 50$。
2	24	兩個世界中的道路全部可以連接成一條路徑。
3	38	平行世界中的每一個里也都可以從編號為 1 的里到達。
4	30	無額外限制

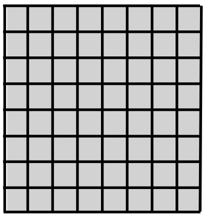
壓縮影像編輯 **(Compressed Image Editing)**

問題描述

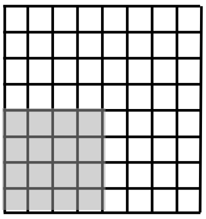
　DF-expression (depth-first picture expression) 是一種壓縮黑白影像的方法。假設影像大小為 n × n，其中 n 是 2 的冪次，DF-expression 的遞迴定義如下。如果每一格像素都是白色，我們用 0 來表式 (如圖 (a))；如果每一格像素都是黑色，我們用 1 來表式 (如圖 (b))；如果並非每一格像素都同色，我們先將影像等分為左上、右上、左下、右下四塊後，然後表示如下: 先寫下 2 ，之後依續接上左上、右上、左下、右下四塊的表示法。(如圖 (c) 和 (d))



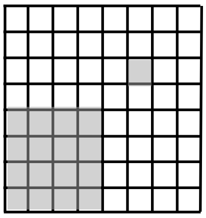
(a) 0



(b) 1



(c) 20010



(d) 2020020100010

　影像經壓縮後，在編輯時會比較麻煩。在這個問題中，我們將給你一張壓縮後的影像 M，以及一個編輯指令 (r1, c1, r2, c2, b)，其中 0 ≤ r1, c1, r2, c2 ≤ n − 1 且 b = 0 或 1，表示要將影像 M 中一個給定矩形區域內的像素全部設為 b (0 表示白色；1 表示黑色)，待編輯的矩形區域由(r1, c1, r2, c2)描述。(r1, c1)代表矩形區域之左上角座標在第 r1 列第 c1 行，(r2, c2)代表矩形區域之右下角座標在第 r2 列第 c2 行。

　舉例來說: 對圖 (a) 下編輯指令 (4, 0, 7, 3, 1) 就會得到圖 (c)；對圖 (c) 下編輯指令 (3, 0, 7, 4, 0) 就會得到圖 (a)。編輯區域的左上角及右下角有可能重合，例如: 對圖 (c) 下編輯指令 (2, 5, 2, 5, 1) 就會得到圖 (d)；相反的，對圖 (d) 下編輯指令 (2, 5, 2, 5, 0) 就會得到圖 (c)。請注意: 如果對圖 (d) 下編輯指令 (2, 5, 2, 5, 1)，影像不會有任何改變。

輸入格式

1. 輸入第一行為兩個正整數 n ($1 \le n \le 10^9$) 和 L ($1 \le L \le 50$)，代表影像 M 的大小為 n × n，DF-expression 的長度為 L，其中 n 必為 2 的冪次。
2. 第二行是影像 M 的 DF-expression (由連續的 0、1、2 字元組成)。
3. 第三行包含五個正整數 r1、c1、r2、c2、b ($0 \le r1 \le r2 \le n-1$，$0 \le c1 \le c2 \le n-1$，b = 0 或 1)，代表一個對 M 的編輯指令。並且保證編輯指令滿足 $(r2 - r1 + 1) \times (c2 - c1 + 1) \le 10^2$，即修改的矩形區域的總點數 ≤ 10^2。
4. 同一行的數值間以一個以上的空白隔開。

輸出格式

請輸出一行代表影像 M 經編輯後的 DF-expression。

輸入範例一	輸出範例一
8 5 20010 2 5 2 5 1	2020020100010
輸入範例二	**輸出範例二**
8 13 2020020100010 2 5 2 5 0	20010
輸入範例三	**輸出範例三**
8 5 20010 3 0 7 4 0	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	10	$n = 2$; $(r2 - r1 + 1) \times (c2 - c1 + 1) = 1$
2	25	$1 \le n \le 10^3$; $(r2 - r1 + 1) \times (c2 - c1 + 1) = 1$
3	30	$1 \le n \le 10^9$; $(r2 - r1 + 1) \times (c2 - c1 + 1) = 1$
3	35	$1 \le n \le 10^9$

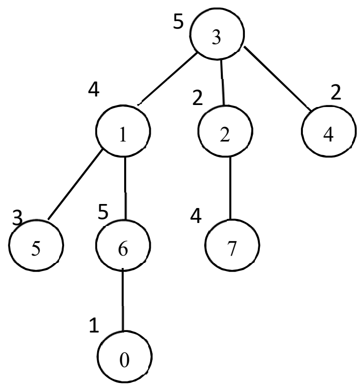
第三題：病毒演化 (Virus)

問題敘述

有一種演化非常快速的病毒，一開始的時候只有原型病毒，原型病毒演化成若干種變異型，每一種變異型又演化成其他的變異型，已知病毒演化的過程中，每次產生的變異型都不相同，因此，整個演化的過程可以用一棵演化樹來表示：原型病毒就是樹根，每一個型態的病毒都是一個樹的節點，親代病毒會演化出一個或多個子代，但也有些病毒不再演化。每個節點都有恰有一條從根節點走到他的路徑，這條路徑上的所有節點(除了他自己)都稱為他的祖先，而每一病毒都是他的祖先的後代。

科學家在研究病毒的某一個特性值，節點 v 的特性值以 $f(v)$ 表示。科學家定義：

若 p 是 v 的祖先，且從 p 到 v 的演化路徑上的每一種病毒(包含 v 但不含 p)的特性值都嚴格小於 $f(p)$，則 v 稱為 p 的**可控制類型**，而 p 的可控制類型的數量就稱為 p 的**可控制量** $c(p)$。此外，科學家們也定義了任何病毒都屬於自己的可控制類型。



根據輸入的演化樹以及各節點的特性值，一種病毒的**可控制總指標**，便定義為原型病毒與所有演化出來病毒的可控制量的總和。以右圖舉例來說，圖中每個圓圈代表一個型態的病毒，也是演化樹的一個節點，圓圈內是節點編號，圓圈外標示的數字是他的特性值。根節點 $root = 3$ 是代表原型病毒，他的可控制類型包括編號 3, 1, 2, 4, 5 與 7，可控制量 $c(3) = 6$。節點 1 的可控制類型是節點 1, 5，$c(1) = 2$；節點 6 的可控制類型是節點 6, 0，$c(6) = 2$；而其他五個節點的可控制類型都只有自己。所以這一範例的可控制總指標等於 $c(3) + c(1) + c(6) + 5\times1 = 15$。

病毒的演化速度實在是太快了，在科學家進行研究的過程中，許多變異型病毒會不斷地產生新的變異。科學家們忙於將變異得到的病毒特性值記錄下來，卻來不及計算新的可控制總指標之值。你能夠協助隨時更新可控制總指標之值嗎？

輸入格式

輸入第一列為一個正整數 $N\,(1 \le N \le 200000)$，代表目前的演化樹之節點數，節點編號為 $0, 1, 2, \ldots, N-1$，第二列是 N 個整數，依序代表 $f(0), f(1), \ldots, f(N-1)$。接下來有 $N-1$ 列表示親子關係，每一列依序出現兩個整數 v 與 p，代表 v 的親代是 p。同一行的數值間以空白隔開，特性值都是不超過 10^8 的非負整數。

接下來有一個正整數 $Q\,(0 \le Q \le 200000)$，代表隨時間新增的變異型病毒。緊接著有 Q 列

數字：其中第 i 列包含兩個數字 p $(0 \le p < N+i-1)$與 f $(0 \le f \le 10^8)$，代表編號 $N+i-1$ 的變異型病毒的親代病毒編號與該病毒的特性值。

輸出格式

對於每一筆測試資料，輸出 Q+1 列，第 i 列包含考慮編號 0 到 $N+i-2$ 病毒組成演化樹的可控制總指標之值。

輸入範例 1	輸出範例 1
8 1 4 2 5 2 3 5 4 1 3 2 3 4 3 5 1 6 1 7 2 0 6 2 0 100 0 200	15 16 17

評分說明

本題共有 5 個子任務，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	9	$1 \le N \le 1000$、$Q = 0$
2	11	每個節點最多只有一個子節點
3	26	$1 \le N \le 200000$、$Q = 0$
4	18	$1 \le N \le 1000$、$0 \le Q \le 1000$
5	36	無額外限制

第四題：籌碼投注 (Bidding)

問題敘述

愛麗絲和鮑伯兩人在玩一個籌碼遊戲，遊戲規則如下，一開始兩人先抽出 7 個正整數，分別為 A, B, N, p, q, r, s，其中 A 代表愛麗絲分配到的籌碼，而 B 代表鮑伯分配到的籌碼，接著兩人輪流投注籌碼到桌面上，從愛麗絲先開始。當輪到愛麗絲投注時，如果愛麗絲有至少 q 枚籌碼，愛麗絲必須投注 $px+q$ 枚籌碼到桌面上(其中 x 是任意非負整數)，否則愛麗絲沒有合法投注方式，愛麗絲只能略過該局。如果該局結束後，桌面上至少有 N 枚籌碼，則愛麗絲贏得遊戲，遊戲結束。當輪到鮑伯投注時，如果鮑伯有至少 s 枚籌碼，鮑伯必須投注 $ry+s$ 個籌碼到桌面上(其中 y 也是任意非負整數)，否則鮑伯沒有合法投注方式，鮑伯只能略過該局。如果該局結束後，桌面上至少有 N 枚籌碼，則鮑伯贏得遊戲，遊戲結束。如果兩人始終都無法贏得遊戲，則判定雙方平手。

現在請你撰寫一支程式，模擬若干場籌碼遊戲，假設兩人都使用最佳的策略投注籌碼的情形下，輸出遊戲的結果。備註：所謂的最佳策略，我們定義為「優先讓自己獲勝，若發現自己無法獲勝，則千方百計不讓對方獲勝」。

輸入格式

輸入的第一列包含一個正整數 $T (1 \le T \le 100)$，代表模擬籌碼遊戲的場數。接下來的 T 列每一列包含七個整數 A, B, N, p, q, r, s。

輸出格式

對於每一場籌碼遊戲，如果愛麗絲有必勝策略請輸出 “ALICE”，如果鮑伯有必勝策略請輸出 “BOB”，如果雙方平手請輸出 “TIE”。(注意：輸出的英文字須全部大寫)

輸入範例 1	**輸出範例 1**
3 3 3 4 2 1 2 2 4 4 4 2 1 2 2 2 3 5 2 1 2 2	ALICE BOB TIE

評分說明

本題共有 5 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	15	$1 \le A, B, N \le 10^9$、$p = 2$、$q = 1$、$r = 2$、$s = 2$
2	11	$1 \le A, B, N \le 100$、$0 \le p, r \le 10$、$1 \le q, s \le 10$
3	12	$1 \le A, B, N \le 1000$、$0 \le p, r \le 1000$、$1 \le q, s \le 1000$
4	20	$1 \le A, B, N \le 10^9$、$0 \le p, r \le 1000$、$1 \le q, s \le 1000$ $A + B \ge N + \max(q, s)$
5	14	$1 \le A, B, N \le 10^6$、$0 \le p, r \le 1000$、$1 \le q, s \le 1000$
6	28	$1 \le A, B, N \le 10^9$、$0 \le p, r \le 1000$、$1 \le q, s \le 1000$

第一題：吠市數列 (Fibonacci)

問題敘述

街上有很多隻流浪狗正在聚集。只要有一隻流浪狗開始吠叫，被影響到的流浪狗就會加入跟著一直吠叫。被影響到幾乎失眠的你，只好放棄數羊這件容易睡著的事情，改成數正在吠叫的流浪狗數量。經過了無數個失眠的夜晚，你觀察到兩個不可思議的現象：其一、如果有一隻流浪狗從第 n 分鐘開始吠叫，那麼只要這隻流浪狗還留在街上，接下來的每一分鐘都會持續吠叫。其二、如果你在第 n 分鐘觀察到恰好有 G_n 隻流浪狗正在吠叫，那麼從第 $n+2$ 分鐘的起，就會額外多出 G_n+1 隻流浪狗一起加入吠叫的行列。

舉例來說，如果從第 0 分鐘開始有 $G_0=3$ 隻流浪狗在叫、而第 1 分鐘開始有 $G_1=4$ 隻流浪狗在叫，那麼上述觀察，我們可以得出 $G_2=4+(3+1)=8$。如果多計算一些，我們可以預測出以下的表格：

n	0	1	2	3	4	5	6	7	8
G_n	3	4	8	13	22	36	59	96	156

你有一位熱心的鄰居小 P，每天晚上只要吠叫的流浪狗數量至少有 P 隻，那麼在同一分鐘內小 P 會找來大量環保局的車子，將這些流浪狗分批載走。每一台環保局的車總是恰好載 P 隻流浪狗離開現場，不多也不少。因此你觀察到的流浪狗數量總是 G_n 除以 P 的餘數。以上面的情境為例，如果 $P=5$，那麼你實際觀察到吠叫中的流浪狗數量，可以寫成以下表格：

n	0	1	2	3	4	5	6	7	8
$G_n \bmod P$	3	4	3	3	2	1	4	1	1

今天又是一個熱鬧的夜晚。數度失眠的你，閒來無事在第 i 分鐘和第 j 分鐘時，在紙上記錄下 $G_i \bmod P$ 以及 $G_j \bmod P$。現在你想要寫個程式來預測出第 k 分鐘的時候會記錄到多少流浪狗在叫。由於你是在失眠的狀態下紀錄這些數字的，如果不存在任何的 G_0 和 G_1 滿足你的觀察，那麼你的程式必須要能夠判斷出此情況並且輸出無解。

輸入格式

每筆測試資料的第一列有一個數字 $T\,(1 \le T \le 10)$，代表失眠的日子數。接下來的 T 列每一列有 6 個非負整數，依序為 P、i、$G_i \bmod P$、j、$G_j \bmod P$、k。

輸出格式

對於每一個失眠的日子，如果你能夠唯一確定 $G_k \bmod P$ 的值，請輸出該值於一列。若無解，請輸出 **-1**。若有超過一個可能的答案，請輸出 **-2**。

輸入範例	輸出範例
4 5 4 2 5 1 6 3 2 1 4 0 6 5 9 0 4 1 6 5 9 2 4 3 6	4 0 -2 -1

評分說明

本題共有 5 個子任務，條件限制如下所示。每個子任務可能有一筆或多筆測試資料，該子任務所有測試資料皆須答對才會獲得該子任務的分數。

子任務	分數	額外輸入限制
1	5	$2 \le P < 2^{31}$；$i = 0$；$j = 1$；$2 \le k \le 20$。P 是質數。
2	13	$2 \le P < 100$；$i = 0$；$j = 1$；$2 \le k < 2^{63}$。P 是質數。
3	21	$2 \le P < 2^{31}$；$i = 0$；$j = 1$；$2 \le k < 2^{63}$。P 是質數。
4	27	$2 \le P < 2^{31}$；$0 \le i, j, k < 2^{63}$。P 是質數。i, j, k 三者相異。 $(G_0 \bmod P, G_1 \bmod P)$ 有唯一解。
5	34	$2 \le P < 2^{31}$；$0 \le i, j, k < 2^{63}$。P 是質數。i, j, k 三者相異。

第二題：記憶體 (Memory)

問題敘述

有 N 個資料檔案，編號為 i 的資料檔案大小為 w_i。資料檔案的編號為 $1, 2, \ldots, N$。

現在有 $2N$-3 個階段的工作需要被**依序執行**：必須要完成一個工作階段，才能夠開始執行下一個工作階段。對於每一個工作階段，它都有指定使用的資料檔案們。在執行工作的過程中，所有指定使用的資料檔案必須要隨時、完整地出現在記憶體中，才能使程式順利執行。在每一工作階段執行後，你可以選擇完全刪除某些資料檔案，騰出更多空間用來載入之後可能使用的資料檔。

每一個工作階段恰好需要存取兩個資料檔案：

- 對於第 $1 \le k \le N$-1 個工作階段，它需要的資料檔案編號恰好為 k 以及 $k+1$。
  也就是說，這些工作依序需要編號為 $\{1, 2\}$、$\{2, 3\}$、…、$\{N\text{-}1, N\}$ 這些資料檔案。
- 對於第 $N \le k \le 2N$-3 個工作階段，它需要的資料檔案編號恰好為 $2N\text{-}1\text{-}k$ 以及 $2N\text{-}2\text{-}k$。
  也就是說，這些工作依序需要編號為 $\{N\text{-}1, N\text{-}2\}$、$\{N\text{-}2, N\text{-}3\}$、…、$\{2, 1\}$ 等資料。

記憶體有容量上限 M。在執行第一件工作之前，記憶體是空的，而所有的資料檔案都被儲存在網路硬碟裡面。把一個大小為 x 的資料檔案搬入記憶體需要花費 x 單位成本。把一個檔案從記憶體刪除則不花費任何成本。請注意，若決定將某個資料檔案從記憶體中刪除，則整個檔案都會被刪除。

現在給定每一筆資料大小，請計算完成所有工作所需要的最小總花費。

輸入格式

輸入的第一列包含兩個正整數 N、M，依序代表資料檔案的數量與記憶體大小。接下來的第二列有 N 個以空白隔開的正整數 $w_1, w_2, \ldots, w_n$。輸入保證對任意 $1 \le k \le N$-1，都有 $w_k + w_{k+1} \le M$。

輸出格式

請輸出依序完成所有工作所需要的最小總花費。

輸入範例 1	輸出範例 1
4 9 5 4 1 6	25

輸入範例 2	輸出範例 2
10 10 1 1 1 1 1 1 1 1 1 1	10

輸入範例 3	輸出範例 3
10 2 1 1 1 1 1 1 1 1 1 1	18

評分說明

本題共有 3 個子任務，條件限制如下所示。每一子任務含有多筆測試資料，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	16	$1 \le N \le 20$，$1 \le M \le 1000$，$1 \le w_i \le 300$。
2	31	$1 \le N \le 50$，$1 \le M \le 7000$，$1 \le w_i \le 500$。
3	53	$1 \le N \le 500$，$1 \le M \le 100000$，$1 \le w_i \le 1000$。

第三題：馬力的比賽 (Beside Mario)

問題敘述

馬力是一位非常熱衷比賽的選手，他每一場比賽都想要參加。馬力最近蒐集了未來一段時間的所有比賽的開始與結束時間，由於這些比賽都規定必須全程參加不可以中途離開，也不可以遲到或早退，所以任兩場比賽的時間如果有重疊就不能都參加。馬力想要規劃參加哪些比賽以便比賽場次越多越好，但是他同時想到另外兩個問題：第一個問題是有多少場比賽是「不可或缺」的，一個比賽被馬力稱為「不可或缺」的意思是指：這場比賽如果被取消，他能參加的最多場次數會因此減少。第二個問題是能參加最多場次的組合有多少種。

第 i 場比賽的時間由一個區間 $T(i) = [s(i), f(i)]$表示，其中 $s(i) < f(i)$ 為兩個非負整數，兩場比賽的區間如果重疊，則不可同時參加。注意，這裡重疊的定義為兩個區間的交集超過一點，也就是說，僅在某一端點相交兩區間是可以同時參加的，例如 $[1, 3]$ 與 $[3, 4]$。

陸堯是馬力的知音，也是好朋友，決定要在馬力出賽的時候看直播遠距離加油。陸堯深知馬力會選擇參加最多的比賽，但除了「不可或缺」的比賽以外，他並不知道馬力會參加哪些比賽。對此，陸堯提出了第三個問題：究竟有總長度多少的時間，能夠保證馬力一定會**正在參加**某場比賽呢？由於直播時間必須要事先設定，陸堯得在比賽開始前就決定好要收看哪些時段的直播，不能中途變更收看直播的時間。

請你寫個程式回答馬力和陸堯的疑問吧！

舉例來說，假設有 6 場比賽的區間是 $T(1) = [0, 3]$、$T(2) = [4, 6]$、$T(3) = [3, 5]$、$T(4) = [7, 9]$、$T(5) = [7, 9]$、$T(6) = [2, 5]$，那麼馬力最多可以參加 3 場比賽，最多場次的組合有 4 種：$\{1, 2, 4\}$、$\{1, 2, 5\}$、$\{1, 3, 4\}$、$\{1, 3, 5\}$，其中第 1 場 $[0, 3]$ 是不可或缺的。此外，無論馬力如何選擇參加 3 場比賽，陸堯總可以在 $[0, 3], [4, 5], [7, 9]$ 這些時段看到馬力的直播，因此總時間是 6 單位長。

輸入格式

輸入第一列為一個正整數 $N\,(1 \le N \le 500000)$，代表比賽的場數，接下來有 N 列，每列包含兩個整數 $s(i)$ 與 $f(i)$，依序代表一場比賽的開始時間與結束時間，同一行的數值間以空白隔開，時間都是不超過 10^8 的非負整數。

輸出格式

對於每一筆測試資料，請輸出三列：第一列為不可或缺的比賽場數，第二列為最多場次的組合數除以 10^9+7 的餘數，第三列輸出保證能看到正在比賽的馬力的總時間長度。

輸入範例 1	輸出範例 1
6 0 3 4 6 3 5 7 9 7 9 2 5	1 4 6

輸入範例 2	輸出範例 2
6 0 5 0 4 0 3 3 7 4 8 5 8	0 6 5

輸入範例 3	輸出範例 3
6 0 1 1 2 2 3 3 4 4 5 5 6	6 1 6

評分說明

本題共有 3 個子任務，條件限制如下所示。每一組可有一或多筆測試資料，對於每一個子任務，若你正確回答出所有測試資料的第一個問題，可以獲得 30% 的得分；正確回答出第二個問題，可以獲得 40% 的得分；正確回答出第三個問題，可以獲得 30% 的得分。

子任務	分數	額外輸入限制
1	9	$N \le 10000$，只有一種最多場次的組合。
2	12	$N \le 10000$
3	79	無額外限制

第四題：離不開的新手村 (Stay)

問題敘述

大家有玩過大風吹嗎？就是那種一開始大家不曉得為什麼很整齊地坐在一塊長 H 寬 W 總共有 HW 個座位的地方，然後一聲令下就不斷地有某兩個人交換位置的莫名其妙遊戲。

在這款名為大風吹的線上遊戲裡面，有 N 座島嶼，以及 M 個傳送通道。每一個傳送通道 (U_i, V_i) 都能夠將玩家**單方向地**從編號為 U_i 的島嶼傳送至編號為 Vi 的島嶼。每一座島嶼上，都有一個大風吹的益智關卡（細節不是很重要我們就不提了）。對於每一個關卡，遊戲公司可以設定其難度。為了讓遊戲變得好玩，遊戲公司決定要將每一個關卡設定一個介於 1 到 N 之間的難度值，而且每一個關卡的難度都不同。

每一個玩家都有一個等級 $L\,(1 \le L \le N)$，如果 L 值小於關卡的難度值，那麼玩家很可能會無法通關。遊戲公司不希望任何新手玩家遇到這樣的情形，一旦玩家無法通關，很可能就永遠不會再登入這款遊戲了。於是，對於一位新加入的等級 L 玩家，遊戲公司隨機決定一個難度值介於 1 至 L 的關卡，並且把該玩家丟到擁有該難度值關卡的島嶼上，確保他們能夠過關。

遊戲公司並不會更動島嶼和傳送通道的內容。不過，為了讓遊戲變得有變化性，每一陣子遊戲公司會更新關卡內容：**交換**兩座島嶼上的關卡難度。如此一來，對於熟悉地圖與傳送通道的玩家們，也能夠有興致在同一座島嶼上重複闖關。

遊戲公司隱約察覺到一件事情：如果玩家被傳送到一個擁有超過自己目前等級關卡的島嶼，這位玩家也會備感沮喪。因此，遊戲公司需要你的協助，對於每一次遊戲更新，統計有多少個新手等級 L，使得玩家從任何一個關卡等級不超過 L 的島嶼出發，經過任意次傳送，都不會被傳送到關卡等級超過 L 的島嶼。

輸入格式

輸入的第一列兩個正整數 N, M $(1 \le N, M \le 500000)$。第二列有 N 個數字，第 i 個數字代表現在編號為 i 的島嶼上面關卡的難度值。第三列開始有 M 列，每一列有兩個數字 (U_i, V_i) 描述一個傳送通道 $(1 \le U_i, V_i \le N)$。接下來的一列包含一個正整數 Q $(1 \le Q \le 50000)$，代表遊戲公司更新關卡內容的次數。緊接著的 Q 列，每一列包含兩個數字 $a, b\,(1 \le a, b \le N)$，代表欲將關卡等級為 a 與關卡等級為 b 的兩關卡互相交換。請注意：a 與 b 並非島嶼編號，而是關卡等級。

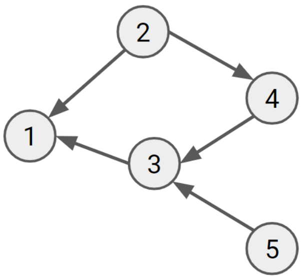
輸出格式

輸出 $Q + 1$ 列。第 i 列$(1 \le i \le Q+1)$包含一個整數，代表在第 $i - 1$ 次更新後，有多少個新

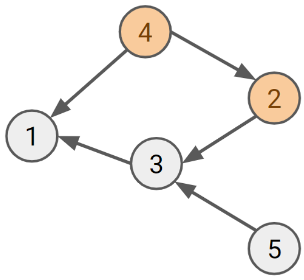
手等級 L 能夠滿足題目要求。

輸入範例 1	**輸出範例 1**
5 5 1 2 3 4 5 2 4 3 1 2 1 4 3 5 3 3 2 4 2 3 5 3	3 4 5 4

一開始有三種可能的新手等級：1, 4, 5。交換等級 2 與等級 4 的關卡以後，可以有四種新手等級：1, 3, 4, 5。**接著**交換等級 2 與等級 3 的關卡以後，可以有五種新手等級：1, 2, 3, 4, 5。**接著**交換等級 5 與等級 3 關卡以後，可以有四種新手等級：1, 2, 3, 5。



初始配置



交換 **(2, 4)**

評分說明

本題共有 6 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	5	$1 \le N, M \le 100$、$1 \le Q \le 100$
2	16	$M = N - 1$、對所有 i，$U_i = i + 1$、$V_i = i$。
3	13	$M = N - 1$、任一島嶼出發總能夠有方法到達編號為 1 的島嶼
4	10	$1 \le Q \le 1000$、$\lvert a - b \rvert \le 10000$
5	25	抵達每一座島嶼的傳送通道數量不超過 10 個 (備註：但可以有超過 10 個傳送通道離開某一座島嶼)
6	31	沒有額外限制

第一題：圓的最佳覆蓋 (Covering)

問題敘述

彼得是一個無線網路公司的員工，最近他在工作上遇到了一個非常困難的問題，因此彼得需要你的幫助。這個問題如下，給定一個整數 $r \geq 1$ 和一個包含平面上 n 個點的集合 $P = \{(x_i, y_i) \mid 1 \leq i \leq n\}$。彼得想要放置一個半徑為 r 的圓，這個圓的圓心可以在平面上任何一個位置，他的目標是讓這個圓蓋住 P 中最多的點。如果一個點出現在圓的邊界上，那麼我們也視同此點被圓所覆蓋。

舉例來說，假設 $P = \{(1, 3), (5, 3), (5, 6), (7, 4), (8, 6)\}$。如果 $r = 3$，最多可以蓋住 P 中 4 個點 (參考圖 1(a))；如果 $r = 2$，最多可以蓋住 P 中 3 個點 (參考圖 1(b))。

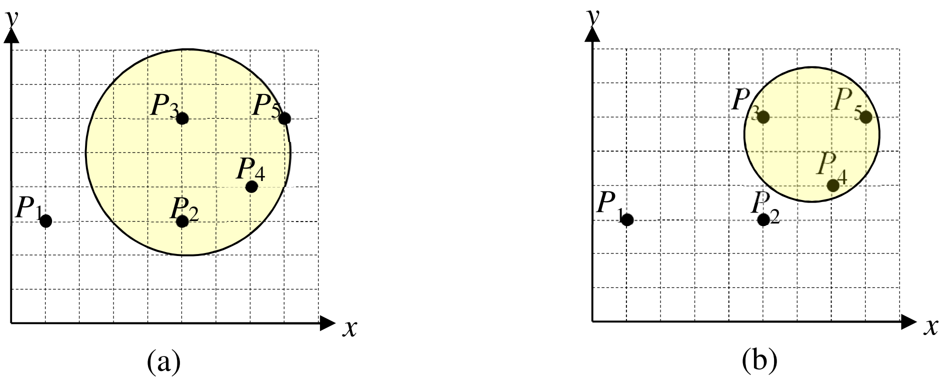


圖 1. (a) 半徑為 3 的圓覆蓋 4 個點; (b) 半徑為 2 的圓覆蓋 3 個點。

給定 r 以及 P 中所有點的座標，請寫一個程式計算一個半徑為 r 的圓最多可以覆蓋 P 中幾個點。注意：此圓的圓心座標可以是任何實數。

輸入格式

每筆測試資料的第一列兩個正整數 n ($1 \leq n \leq 1000$)、以及 r ($1 \leq r \leq 10^4$)，分別代表 P 中座標點的個數，以及圓的半徑。接下來的 n 列，每一列包含兩個正整數 x, y ($-10^4 \leq x, y \leq 10^4$)。

輸出格式

請輸出此圓能蓋住 P 中最多點時的點數。

輸入範例 1	輸出範例 1
5 3 1 0 3 0 10 0 7 0 12 0	3

輸入範例 2	輸出範例 2
5 2 1 3 5 3 5 6 7 4 8 6	3

輸入範例 3	輸出範例 3
2 3 1 3 8 6	1

評分說明

本題共有 3 個子任務，條件限制如下所示。每個子任務可能有一筆或多筆測試資料，該子任務所有測試資料皆須答對才會獲得該子任務的分數。

子任務	分數	額外輸入限制
1	8	P 中所有點都在 x 軸上，$1 \leq n \leq 20$。
2	21	$1 \leq n \leq 20$
3	31	$1 \leq n \leq 300$
4	40	$1 \leq n \leq 1000$

第二題：時光寶盒 (Capsule)

問題敘述

鳳凰花開，又到了畢業的季節。樂樂高中三年甲班在離別之際，各自寫下對自己十年後的期許裝進時光寶盒，埋在教室外的榕樹下，約定十年後再次相聚共同開啟時光寶盒，回味高中時光並分享彼此的生活點滴。

為了讓時光寶盒能在十年之後順利被找到，三年甲班的同學決定在學校中庭找一個合適的位置埋下時光寶盒。學校的中庭可以視為一個由 1×1 大小的單位區塊組成的 R 列 C 行矩形土地。時光寶盒是一個長方形的物件，大小可以視為由 $A\times B$ 個單位區塊組成的矩形。可惜的是，中庭的每一單位土地的高度都不盡相同。為了避免時光寶盒傾斜或翻覆，同學們決定挖掉一些土方，使得這個大小為 $A\times B$ 的時光寶盒可以平穩地放在中庭裡。為了表示慎重（也為了讓題目比較好解），時光寶盒在放入中庭時必須要恰好佔據 A 列 B 行，而且要對齊單位區塊的格線。

這個長方形的時光寶盒有一個穩固係數 P。假設該盒子所放置的範圍中，移除土方後土地高度的最大值為 X，那麼對於範圍內任意大小為 $P\times P$ 的矩形區域，至少都要有一個單位區塊的高度也為 X。

給定中庭每一個單位區塊的土地高度，請問至少要移除多少立方單位的土，才能夠平穩地放置這個時光寶盒？

輸入格式

輸入的第一列包含五個正整數 R, C, A, B, P ($1 \le R\times C \le 10^5$、$1 \le A \le R$、$1 \le B \le C$、$1 \le P \le \min(A, B)$)。接下來有 R 列，每一列包含 C 個正整數。第 i 列的第 j 個正整數 $h_{i,j}$ 代表位置 (i, j) 的單位區塊的土地高度($1 \le h_{i,j} \le 10^4$)。

輸出格式

輸出的第一列包含一個正整數，代表需要移除的最小土方總量。

輸入範例 1	**輸出範例 1**
7 7 3 4 2 1 2 3 4 5 6 7 7 6 5 4 3 2 1 1 4 3 5 6 2 7 5 3 4 2 1 6 7 7 4 3 2 6 1 5 4 5 6 1 2 7 3 5 6 4 7 3 2 1	1

輸入範例 2	**輸出範例 2**
3 3 3 3 1 1 1 2 2 3 1 3 2 1	7

輸入範例 3	**輸出範例 3**
2 5 2 3 2 1 1 1 1 1 1 1 2 1 1	0

評分說明

本題共有 6 個子任務，條件限制如下所示。每一子任務含有多筆測試資料，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	5	$A = 1$、$1 \leq h_{i,j} \leq 2$
2	13	$A = 1$
3	14	$P = 1$
4	16	$1 \leq R, C \leq 20$
5	19	$1 \leq h_{i,j} \leq 20$
6	33	無額外限制

第三題：網路佈線問題 (Network)

問題敘述

小明在天龍市的資訊處打工，協助市內網路光纖佈線的業務，最近他在幫一社區規劃新一代的網路架構。為方便起見，以 1, 2, …, N 表示各網路設備的編號，並以 1 表示總機房的設備，若設備 i 與設備 j（$1 \le i, j \le N$）之間有直接連接的線路管溝時，令 $A_{i,j}$ 表示它們之間的距離，兩個相鄰網路設備可以透過管溝鋪設光纖直接連結起來。不相鄰的網路設備則可經由幾個設備連結起來，其距離為所經過的管溝距離總和。

小明希望能使用最少的光纖（也就是少任何一條光纖都會使得某兩個設備無法直接或間接連結)、與盡量短的光纖（也就是光纖總長度盡量短）將各個網路設備連結起來，另外小明也希望，在查線時沿著所佈的光纖由總機房走到各個網路設備的距離越短越好。

給定兩個正整數 a 和 b，我們令一個實數參數 $K = a/b$。給定一網路架構圖，定義 dist(i) 代表按照所規劃網路佈線的方式，由設備 1 到設備 i 的距離長度，並定義 dist′(i) 代表把所有管溝全部都鋪設光纖的情境下，由設備 1 到設備 i 的最短路徑長度。令能將所有網路設備直接或連接起來所需要的最短光纖總長度為 MST，請寫一程式幫小明規劃網路佈線的方式，滿足以下三個條件：

1. 實際佈線光纖數量恰好為 N-1 條；
2. 實際佈線光纖總長度不超過 $(1+2/K)MST$；
3. 對所有 i 必須滿足 $\text{dist}(i) \le (1+K)\ \text{dist}'(i)$。

你可以假設，輸入所提供之管溝資訊，必定存在一種滿足上述條件的佈線方式。

輸入格式

輸入的第一列包含四個正整數 N, M, a, b ($2 \le N \le 2048$、$N\text{-}1 \le M \le 10^5$、$1 \le a, b \le 10$)，其中 N 與 M 分別代表網路設備的數量與可鋪設光纖的管溝數量。其中 a, b 則用來定義問題敘述裡的參數。

接下來有 M 列，每一列有三個正整數 i, j 和 $A_{i,j}$，代表設備 i 與設備 j 的管溝距離為 $A_{i,j}$。所有輸入的數值都不超過 10^6。輸入的管溝保證能夠讓所有的設備直接或間接連在一起。任何兩個設備之間至多只有一個管溝。

輸出格式

輸出 N-1 列，代表任何一種滿足條件的網路佈線方式：每一列包含兩個正整數 i, j，表示要在設備 i 與設備 j 之間的管溝佈線。

輸入範例 1	輸出範例 1
3 3 1 1 1 2 1 1 3 5 2 3 1	1 2 2 3

輸入範例 2	輸出範例 2
4 5 1 10 1 2 2 1 3 2 1 4 2 2 3 1 3 4 1	1 2 1 3 1 4

評分說明

本題共有 4 個子任務，條件限制如下所示。每個子任務可能有一筆或多筆測試資料，該子任務所有測試資料皆須答對才會獲得該子任務的分數。

子任務	分數	額外輸入限制
1	10	$N = 10$，$M = N$，每個設備直接與另外兩個設備相連。
2	25	$2 \le N \le 50$，$N\text{-}1 \le M \le 100$，最多只有兩種管溝的長度。
3	39	$2 \le N \le 256$，$N\text{-}1 \le M \le 1000$。
4	26	無額外限制

第四題：自助餐 (Buffet)

問題敘述

選訓營的同學每一天都要吃自助餐。每一天，餐廳裡面都提供了一模一樣的 m 道菜可以選擇，但因為預算有限的關係，助教只能幫同學們挑選**恰好** k 道菜打包成一個便當。

為了讓同學能均衡攝取營養，助教決定對於任何 m 道菜中的 d+1 道菜($d+1 \le k$)組合，只能在選訓營舉辦期間出現在便當**至多一次**。為了能讓選訓營舉辦越多天越好，你決定要幫助教寫一支程式產生出盡量多的便當組合。

輸入格式

輸入的第一列有五個正整數 m, k, d, N_{50}, N_{100}，其中 m, k, d 之定義請參考問題敘述，而 N_{50} 與 N_{100} 之定義請參考評分說明。這個題目其實是由 Output Only 題目偽裝而成的，請查閱評分說明以取得所有測試資料的輸入值。

輸出格式

請於第一列輸出一個正整數 N，代表你輸出的便當數量。接下來的 N 列，每一列都輸出一個長度恰好為 m 的 0-1 字串，若第 i 列的第 j 個字元為 1，代表第 i 天的便當包含了第 j 道菜。

輸入範例 1	**輸出範例 1**
5 3 2 3 5	5 00111 01011 10011 10101 11100

評分說明

本題共有 9 個子任務，條件限制如下所示。若你的輸出不滿足題目敘述之要求，則得分為 0。若你的輸出滿足題目敘述之要求，而該子任務所佔分數為 X，此時得分規則如下：

- 如果輸出的 N 滿足 $N < N_{50}$ 或 $N > N_{100}$，那麼得分為 0。
- 如果輸出的 N 滿足 $N_{50} \le N < N_{100}$，那麼你將獲得 $\left(0.5 + 0.5\frac{N\text{-}N_{50}}{N_{100}\text{-}N_{50}}\right)X$ 分。
- 如果輸出的 $N = N_{100}$，那麼你可以獲得 X 分。

子任務	分數	輸入限制
1	3	$m = 5, k = 3, d = 2, N_{50} = 5, N_{100} = 10$
2	6	$m = 8, k = 5, d = 3, N_{50} = 7, N_{100} = 8$
3	10	$m = 20, k = 12, d = 7, N_{50} = 12, N_{100} = 16$
4	12	$m = 64, k = 32, d = 16, N_{50} = 1, N_{100} = 125$
5	13	$m = 65, k = 32, d = 17, N_{50} = 50, N_{100} = 125$
6	13	$m = 49, k = 7, d = 3, N_{50} = 1200, N_{100} = 2401$
7	14	$m = 121, k = 11, d = 4, N_{50} = 3600, N_{100} = 161051$
8	14	$m = 1369, k = 37, d = 2, N_{50} = 7200, N_{100} = 50653$
9	15	$m = 49, k = 7, d = 5, N_{50} = 14400, N_{100} = 117649$

第一題：再生 (Reborn)

問題敘述

久遠久遠以前，存在一個由正整數所構成的國度，在這個國度裡面，名為「再生」的程序不斷地進行著。所謂的再生，是指國度中最大的兩個正整數進行的融合與分裂後轉變為三個新正整數的過程。精確來說，這兩個最大的數會先融合為此二數之和 S，之後 S 會分裂成三個數字，滿足三數之和為 S 且任兩數最多差 1。若國度中最大的數不止兩個，則依然只有其中某二數會進行再生的程序。然而，再生並不會隨意進行；進行的充分必要條件是在國度中有兩個數字之和大於等於 K，其中 K 為「再生門檻」，為一大於等於 3 的數。對於一個不會發生再生的國度，我們稱其為「穩定國度」。

比如說，考慮一由三數 4, 6, 9 構成的國度；若 $K = 9$，則再生會於此國度進行，首先 6 與 9 融合為 15，之後 15 分裂成 5, 5, 5 三個數字，因此經歷完第一次再生後，此國度的成員會變為 4, 5, 5, 5 四個數字。我們可以很容易的檢測，當前的國度並非穩定國度，故會進行第二次再生；在第二次再生過程中，最大的二數 (某兩個 5) 會融合成 10，之後分裂成 3, 3, 4；在第二次再生完成後，國度中有 3, 3, 4, 4, 5 五個數字。這樣的再生會不斷地進行，直到國度變為穩定國度（再生終會停止，這是顯而易見的）。以此範例來說，再生停止時國度中的數字為 3, 3, 3, 3, 3, 4。

今針對一給定的國度，請計算此國度需要經歷幾次的再生，才會變為「穩定國度」。

輸入格式

每筆測試資料的第一列有兩個正整數 n, K ($2 \le n \le 10^5$、$3 \le K \le 10^{12}$)，分別代表一國度最初的正整數個數以及該國度的再生門檻。接下來的 n 列，每一列有一個正整數，這 n 個數字代表了國度中一開始的數字。所有數字都不會超過 10^{12}。

輸出格式

在單一行中，輸出此國度須經歷的再生總數。

輸入範例 1	輸出範例 1
2 3 3 2	3

輸入範例 2	輸出範例 2
3 3 999 223 69095	70314

輸入範例 3	輸出範例 3
2 10 8 7	2

評分說明

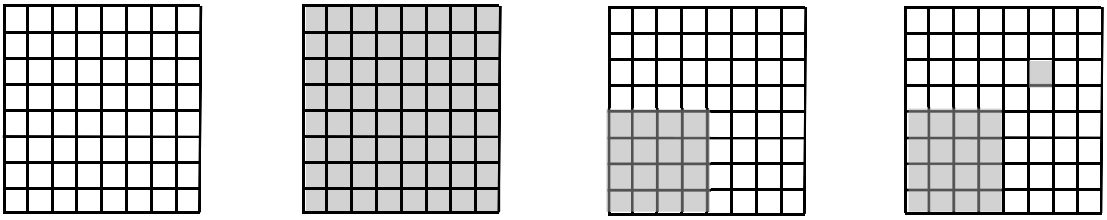
本題共有 4 個子任務，條件限制如下所示。每個子任務可能有一筆或多筆測試資料，該子任務所有測試資料皆須答對才會獲得該子任務的分數。

子任務	分數	額外輸入限制
1	9	$2 \le n \le 10$、$3 \le K \le 10$、所有數字皆不超過 100。
2	17	$K = 3$。
3	18	$2 \le n \le 10^3$、所有數字皆不超過 10^6。
4	23	$2 \le n \le 10^4$、所有數字皆不超過 10^9。
5	33	無額外限制。

第二題：拆解壓縮影像 (Decomposition)

問題敘述

DF-expression (depth-first picture expression) 是一種壓縮黑白影像的方法。假設影像大小為 $n \times n$，其中 n 是 2 的冪次，DF-expression 的遞迴定義如下：如果每一格像素都是白色，我們用 0 來表示 (如圖 (a))；如果每一格像素都是黑色，我們用 1 來表示 (如圖 (b))；如果並非每一格像素都同色，我們先將影像等分為左上、右上、左下、右下四塊後，然後表示如下：先寫下 2，之後依續接上左上、右上、左下、右下四塊的表示法。(如圖 (c) 和 (d))



(a) n=8, “0” (b) n=8, “1” (c) n=8, “20010” (d) n=8, “2020020100010”

影像經過壓縮後，一些常見的演算法執行起來就會變得困難許多。在這個問題中，我們將給你一張壓縮後的影像 **X**。你的任務是要將黑色像素形成的連通區域分離出來 (視為一張單一的影像)，並且分別找出每一個連通區域長度最短的 DF-expression。請注意，如果兩個像素僅有角落接觸到，它們不算是直接連通的。為了方便起見，你只需要由小到大輸出這些 DF-expression 的長度就可以了。請注意，對於每一個連通區域你必須分別挑選最恰當的 2 的冪次作為影像邊長，使得他們在該影像中 DF-expression 字串長度最短。

舉例來說：圖 (b) 只有一個連通區域，可視為一 8×8 的影像，因此最佳壓縮方式就是 “1”。而圖 (c) 也只有一個連通區域，將其視為一 4×4 的影像來壓縮，壓縮後也可以得到 “1”。對於圖 (d) 來說，總共有兩個連通區域，最佳的壓縮方式各自都是 “1”，因此這時候需要輸出兩個 1。

輸入格式

輸入的第一列包含一個正整數 n。第二列有一個字串 S 表示影像 **X** 的 DF-expression。其中 n 必為 2 的冪次。

輸出格式

請於第一列輸出連通區域的數量 k。接下來輸出 k 列請由小到大輸出這些連通塊壓縮後的最短 DF-expression 長度。

輸入範例 1	輸出範例 1
8 2020020100010	2 1 1

輸入範例 2	**輸出範例 2**
8 20010	1 1

輸入範例 3	**輸出範例 3**
8 2200112100021100 20111	3 1 1 5

評分說明

本題共有 5 個子任務，條件限制如下所示。每一子任務含有多筆測試資料，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	7	$1 \le n \le 2^3$，$1 \le \lvert S\rvert \le 85$，影像至多只有一個連通區域。
2	20	$1 \le n \le 2^7$，$1 \le \lvert S\rvert \le 100$，影像至多只有一個連通區域。
3	24	$1 \le n \le 2^{10}$，$1 \le \lvert S\rvert \le 1000$。
4	30	$1 \le n \le 2^{30}$，$1 \le \lvert S\rvert \le 200$，影像至多只有一個連通區域。
5	19	$1 \le n \le 2^{30}$，$1 \le \lvert S\rvert \le 2000$。

第三題：歐拉與 TOT (Totient)

問題敘述

對於任何正整數 n, 歐拉(Euler)的 Totient 函數 φ 定義為：「1 到 n 之間，與 n 互質的整數個數。」舉例來說，1 到 6 之間，與 6 互質的整數有 1 與 5，因此 $\varphi(6) = 2$。

經過推導，可以知道 $\varphi(n)$ 的計算公式為 $\varphi(n) = \prod_{p|n}\left(1-\frac{1}{p}\right)$ 其中連乘符號內的 p 跑遍 n 的所有質因數。

日前，在知名的 TOT 公司所舉辦的數論集訓營裡，學員們接到了一個任務，內容如下：給定一個整數數列 $a_1, a_2, ..., a_n$，學員必須對數列進行 Q 次操作或詢問：

- 乘法操作 "MUL L R x"：將 $a_L, a_{L+1}, ..., a_R$ 這幾個數字的值分別乘上 x。
- TOT 詢問 "TOT L R"：計算並輸出 $\varphi\left(\prod_{i=L}^{R} a_i\right) \bmod (10^9+7)$ 之值。

請你寫一個程式協助 TOT 的學員們完成上述任務。

輸入格式

輸入的第一列兩個正整數 n, Q ($1 \le n \le 4\times10^5$, $1 \le Q \le 2\times10^5$)，分別代表數列的長度、以及進行操作與詢問的總次數。第二列有 n 個正整數 $a_1, a_2, ..., a_n$ ($1 \le a_i \le 300$)，為給定的數列。接下來有 Q 列操作或詢問，每一列格式如題目所述。所有乘法操作的數值 x 均滿足 $1 \le x \le 300$。

輸出格式

對於每一個 TOT 詢問，輸出其所對應的值。

輸入範例 1	輸出範例 1
4 4 5 9 1 2 TOT 3 3 TOT 3 4 MUL 4 4 3 TOT 4 4	1 1 2

輸入範例 2	輸出範例 2
4 4 10 10 10 10 MUL 1 4 5 TOT 1 4 MUL 1 4 2 TOT 1 4	6250000 100000000

評分說明

本題共有 4 個子任務，條件限制如下所示。每個子任務可能有一筆或多筆測試資料，該子任務所有測試資料皆須答對才會獲得該子任務的分數。

子任務	分數	額外輸入限制
1	7	$1 \le n + q \le 10$。
2	11	$1 \le n + q \le 10000$。
3	23	所有操作都是 TOT 詢問。
4	59	無額外限制。

第四題：邊緣的機場 (Outcast)

本題是互動題

問題敘述

因應恐怖攻擊頻傳，為了維護飛航安全，樂樂王國重新檢視了其境內的 N 座機場及相關航線的安全性。一條「航線」，指的是一個航班從某機場 x 直飛至另一個機場 y，中途沒有降落或轉機的路線。航線的安排是單向的，兩座機場間不一定總是會有往返的班機。此外，這 N 座機場編號為 $1, 2, ..., N$。其中編號 1 的機場鄰近首都樂樂城。

樂樂王國的國家安全局正在幫首都樂樂城規劃一系列的安全措施。其首要任務就是要標記出一小群包含編號 **1** 且不超過 $\boldsymbol{N/3}$ 個機場的機場集合，使得從集合內的機場，飛往其他沒被標記的機場的航線總數量不超過 k。

如果這樣的一群機場存在，國家安全局便需要提高警覺與加強安全措施。不過呢，這群機場的機場數量如果很多的話，其實也不用太過擔心。因此，你被指派的任務便是要寫一支程式，試圖找出任何一群滿足條件的機場集合 *result*，或者回報「對於任何大小不超過 m、且包含機場 1 的機場集合 X，其飛往 X 以外機場的航線總數都至少有 $k+1$ 條」。

由於已經多年沒有整理航線的資料，你只能一個一個聯絡其境內的機場，向他們詢問關於由該機場駛出的航線資料。

實作細節

你需要完成以下函式：

```
bool has_outcast_airports(int N, int m, int k,
                          std::vector<int>& result);
```

- N 表示樂樂王國境內有 N 座機場、m 與 k 對應至題目敘述中的條件。
- 如果找得到滿足題目條件的機場集合，請回傳 **true**，並且將該集合內的機場編號不重複地儲存於 *result* 中。*result* 的大小必須介於 1 與 $N/3$ 之間。
- 如果任何大小不超過 m 且包含編號 1 的機場集合，其聯外的飛行航線數量至少為 $k+1$，請回傳 **false**。

你的程式可以呼叫以下兩種函式：

```
int degree(int x);
int get_outgoing_flight(int x, int i);
```

- x 為機場編號，必須是一個介於 1 與 N 之間的整數。
- degree(x) 回傳從機場 x 出發的所有航線總數。
- 針對機場 x，以 x 為出發點的航線編號由 0 至 degree(x) - 1；從同一個機場出發的所有航線目的地機場的編號保證相異。
- get_outgoing_flight(x, i) 回傳從 x 出發的所有航線當中，第 i 條航線的目的地機場編號。參數 i 滿足 $0 \le i \le$ degree(x) -1。
- 對於每筆測試資料，上述兩個函式被呼叫的總次數不能超過 500000 次。

如果不滿足上述條件、或是回傳值不符合題目要求，你的程式會被判為 **Wrong Answer**；否則你的程式會被判斷為 **Accepted**。

互動範例

考慮以下的測試資料：$N = 10$、$m = 2$、$k = 1$。

評分程式呼叫 has_outcast_airports(10, 2, 1, *result*)，一個被評分程式判斷為 Accepted 的互動例子顯示如下：

Call	Return
degree(1)	2
get_outgoing_flight(1, 0)	2
get_outgoing_flight(1, 1)	3
degree(2)	1
get_outgoing_flight(2, 0)	3
degree(3)	2
get_outgoing_flight(3, 0)	2
get_outgoing_flight(3, 1)	4
	true, *result* = [1, 2, 3]

在上面這個例子當中，任何包含機場 1 且大小在 2 以下的機場集合，其聯外的航線總數皆超過 1，故回傳 false 也會被評分程式判定為正確。

評分說明

本題共有 5 個子任務，條件限制如下所示。每個子任務可能有一筆或多筆測試資料，該子任務所有測試資料皆須答對才會獲得該子任務的分數。

子任務	分數	輸入限制
1	4	$N = 10$、$1 \le m \le 3$、$0 \le k \le 3$
2	7	$N = 1000$、$1 \le m \le 10$、$0 \le k \le 10$
3	14	$N = 10^6$、$1 \le m \le 100$、$k = 0$
4	41	$N = 10^6$、$1 \le m \le 100$、$k = 1$
5	34	$N = 10^6$、$1 \le m \le 100$、$0 \le k \le 10$

範例評分程式

範例評分程式以下列格式讀取輸入：

- 第 1 列：N, m, k
- 第 2 列：M
- 第 3 ~ M+2 列：x_i, y_i

其中 N, m, k 如題目所述。M 為航線總數。自第 3 列起每一列的 x_i, y_i 表示有一個航線其從編號 x_i 的機場飛往編號 y_i 的機場。

請注意：使用自己上傳的測試資料進行測試時，沒有下面 MSG 描述的情形時你總會得到 Accepted。如果你的程式被評為 **`Accepted`**，範例評分程式輸出 Accepted: q，其中 q 表示呼叫函式的總次數。如果你的程式被評為 **`Wrong Answer`**，範例評分程式輸出 Wrong Answer: MSG，其中 MSG 格式與意義如下：

- invalid degree query: 不合法的 `degree()` 呼叫。
- invalid flight query: 不合法的 `get_outgoing_flight()` 呼叫。
- too many queries: 呼叫上述兩個函式的總次數超過 500000 次。
- invalid result: 回傳的 *result* 不是一個由 1 到 N 間（包含 1 與 N）不重複的數字構成的集合。
- incorrect result: 回傳的 *result* 不滿足題目條件。