



臺灣大學程式解題社

National Taiwan University Competitive Programming Club

NTUCPC Speed Run! #2

November 9, 2025

Problem	Problem Name	Problem	Problem Name
A	水火箭	N	狙擊對決
B	選舉	O	股票大亨的獲利追蹤器
C	大魔法藥水的祕密配方	P	模運算下的聯立方程式
D	派對	Q	火車運輸系統
E	巫師的多項式挑戰	R	機密竊取
F	假廣告	S	分段時間
G	模仿大師	T	外送系統優化
H	跑操場	U	傳送門
I	圈圈叉叉	V	數學之神
J	牛肉麵店的隊伍	W	水果賽車
K	Git 分支圖	X	平衡之森
L	中樞灌溉系統	Y	畫家
M	太陽系	Z	過場動畫

This page is intentionally left blank.

A. 水火箭

Problem ID: rocket

你是星際水火箭救援小隊 (Intergalactic Water Rocket Rescue Team) 的成員。在一次於水星球 (Planet Aqua) 的測試任務中，你的小隊將一枚水火箭垂直發射至天空中。

由於高度感測器故障，只記錄到火箭到達的最大高度為 h 公尺，之後便開始下墜。水星球的重力與地球相似，向下的加速度為 9.8 m/s^2 。也就是說，火箭上升時，每秒速度會減少 9.8 公尺 直到達到最高點。

為了校準下一次任務的發射系統，你需要計算火箭起飛時的初速度，也就是它剛離開地面時的速度。

Input

輸入僅包含一行一個整數 h ：火箭到達的最大高度（單位為公尺）。

- $0 < h \leq 10^9$

Output

輸出一個浮點數：火箭的初速度（單位為 m/s ）。

當答案的絕對誤差或相對誤差不超過 10^{-6} 時，將視為正確。

Sample Input 1

49	30.99032106965
----	----------------

Sample Output 1

This page is intentionally left blank.

B. 選舉

Problem ID: election

一年一度的水果王國代表選舉在幾天後就要開始了！但是在水果王國的人口指數成長的狀況下，人工計算投票並實時更新已經無法支援今年的選舉了。

水果王國最多有 26 個候選人，每個候選人使用一個大寫英文字母代表，王國的選舉委員會打算使用新的自動化工具支援下面兩種詢問：

- 對於指定的候選人新增一張票。
- 依照字母順序回報所有得票數最多的候選人。

你能幫水果王國完成這個任務嗎？

Input

輸入的第一行只有一個整數 n ($1 \leq n \leq 10^5$) 代表詢問的數量。接下來有 n 行，每一行代表一個詢問，每個詢問都是下面兩個格式之一：

- $+ C$ ：對候選人 C 新增一張票。
- $?$ ：回報所有得票數最多的候選人。這類詢問不會是第一個詢問，並且所有詢問至少出現一次這種詢問。

Output

對於所有「回報」的詢問，分別在一行輸出一個字串，代表所有得票數最多的候選人，按照字母順序排序。一個詢問的兩個輸出字元之間不應有空格。

Sample Input 1

```
9
+ W
+ M
+ K
?
+ W
+ W
+ M
+ K
?
```

Sample Output 1

```
KMW
W
```

Sample Input 2

```
8
+ A
+ B
?
+ A
?
+ B
+ B
?
```

Sample Output 2

```
AB
A
B
```

Sample Input 3

```
6
+ F
+ R
+ U
+ I
+ T
?
```

Sample Output 3

```
FIR TU
```

C. 大魔法藥水的祕密配方

Problem ID: elixir

大魔法藥水是知名的餐廳魔堡王（由謝老闆創立）的招牌商品。

身為其對手餐廳魔之霸（由 P 老闆創立）的密探的你，經過了多年的潛伏，終於成功潛入魔堡王的地下實驗室，偷走了兩瓶製作超級藥水的神秘溶液。你還發現，只要按照一定的比例混合這兩瓶溶液，就能調配出大魔法藥水。

第一瓶溶液含有 $a\%$ 魔法精華，第二瓶含有 $b\%$ 魔法精華。經過精密測量，你觀察出最終調配出的大魔法藥水的含有 $c\%$ 的魔法精華。

請幫 P 老闆算出，要配出恰好 k 公克的大魔法藥水，應該從兩瓶原始溶液中各取多少公克。

魔法精華的濃度定義為：

$$\frac{\text{魔法精華的質量}}{\text{溶液的總質量}}$$

混合規則：

- 混合溶液的總質量等於個別溶液使用的質量之和。
- 混合溶液的魔法精華總質量等於兩瓶溶液的魔法精華質量和。

保證存在唯一解。

Input

一行包含四個以空白隔開的整數 a, b, c, k ：

- a 為第一瓶溶液的魔法精華百分比。
 - b 為第二瓶溶液的魔法精華百分比。
 - c 為最終混合物的魔法精華百分比。
 - k 為要製作的大魔法藥水的總重量（單位為公克）。
-
- $0 \leq a, b, c \leq 100$
 - $a \neq b$
 - $1 \leq k \leq 10^9$

Output

輸出一行包含兩個以空格分隔的非負整數 x 和 y ，其中 x 為第一瓶溶液的重量（單位為公克）， y 為第二瓶溶液的重量（單位為公克）。

保證存在唯一解。

當答案的絕對誤差或相對誤差不超過 10^{-6} 時，將視為正確。

Sample Input 1

Sample Output 1

25 75 35 100	80.0000000000000000 20.0000000000000000
--------------	---

D. 派對

Problem ID: party

NTU 市是一個熱愛運動的城市，城市中的市民個個都是運動好手。作為傳統，NTU 市每年都會舉辦一場全市運動會，來決定誰是全城市最強的運動員。為了慶祝，在運動會中獲得總冠軍的人會舉辦一場派對，並邀請所有他的朋友來參加，同時，他會允許他的朋友邀請任意數量自己的朋友來參加派對。

作為 NTU 市的市民，小李好奇，假設只有慶祝派對的主人和被邀請允許出現的人才會出現在派對，那麼，如果知道全城市中哪些人彼此是朋友，以及全市運動會的總冠軍是誰，最多會有多少人出現在派對上呢？

Input

輸入第一行包含三個正整數 N, M 和 W ，代表 NTU 市的市民總數和城市中，總共有幾對朋友關係，並且第 W 位市民是運動會的總冠軍。

接下來有 M 行輸入，第 i 行包含兩個正整數 x_i, y_i ，代表第 x_i 位市民和第 y_i 位市民是朋友。

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq M \leq 2 \times 10^5$
- $1 \leq x_i, y_i, W \leq N$
- $x_i \neq y_i$
- $\forall 1 \leq i, j \leq M, i \neq j, (x_i, y_i) \neq (x_j, y_j) \text{ and } (x_i, y_i) \neq (y_j, x_j)$

Output

輸出一個正整數，代表派對上最多可能出現的人數。

Sample Input 1

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

Sample Output 1

```
4
```

Sample Input 2

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

Sample Output 2

```
5
```

E. 巫師的多項式挑戰

Problem ID: poly

在魔法王國阿爾格利亞，偉大的巫師波里寧 (PolyNim) 創造了一個強大的多項式咒語：

$$P(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

為了考驗你對多項式魔法的掌握，他將對你問多個問題。每個問題中，給定一個整數 k ，你需要計算多項式 $P(x)$ 在 $x = k$ 處的值。因為計算的結果可能會很大，所以你只需要告訴他結果的後九位，並且，你不需要輸出任何的前導零。舉例來說，如果計算的結果為 1,000,000,001，只需要輸出 1 即可，而如果計算的結果為 1,000,000,000，只需要輸出 0 即可。

Input

第一行包含一個整數 n ，表示多項式的次數。第二行包含 $n + 1$ 個整數 a_0, a_1, \dots, a_n ，表示多項式的係數。第三行包含一個整數 Q ，表示問題的數量。接下來的 Q 行，每行包含一個整數 k ，表示該問題要計算的點。

- $0 \leq n \leq 2 \cdot 10^5$
- 對於所有 $0 \leq i \leq n$ ， $0 \leq a_i \leq 10^9$
- $a_n \neq 0$
- $1 \leq Q \leq 2 \cdot 10^5$
- 每個問題中， $0 \leq k \leq 10^9$
- $n \cdot Q \leq 2 \cdot 10^5$

Output

對每個問題，輸出一行包含一個整數代表 $P(k)$ 不包含前導零的後九位數。

Sample Input 1	Sample Output 1
<pre>2 1 2 3 3 2 0 10</pre>	<pre>17 1 321</pre>

Sample Input 2

```
0
5
5
20
40
27
33
99
```

Sample Output 2

```
5
5
5
5
5
```

Sample Input 3

```
7
7 0 2 90 54 66 38 20
4
0
16
20
6
```

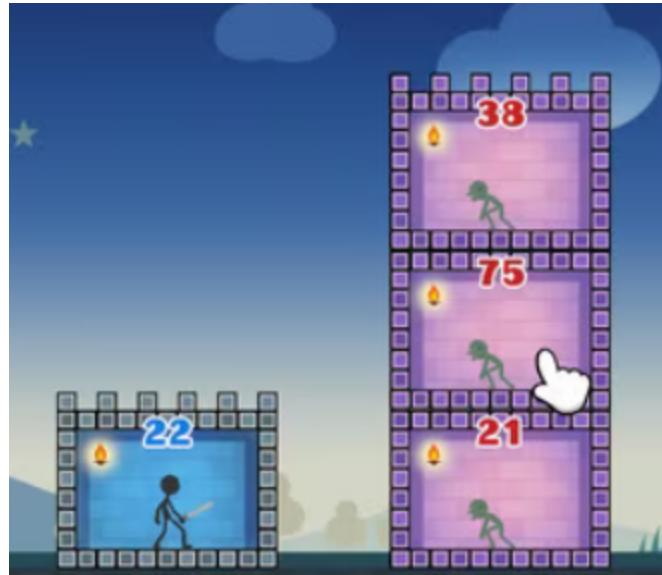
Sample Output 3

```
7
79357447
252560807
7974367
```

F. 假廣告

Problem ID: ad

小明很喜歡玩手機遊戲，不過，最近他發現有些遊戲居然會做跟遊戲內容無關的廣告來吸引人下載，被欺騙的小明很生氣，於是決定把這些假廣告的內容出成程式題目！



小明看到的其中一個廣告如上圖所示，是一個算數爬塔遊戲。遊戲中由左至右一共有 N 座塔，第 i 座塔上會有 k_i 個敵人和 s_i 把寶劍，每位敵人都擁有自己的戰力 $a_{i,j}$ 。玩家一開始會有戰力 X ，並且從最左邊的塔出發。遊戲進行時，玩家可以挑選塔上任何一位尚未選過的敵人戰鬥，並且如果他的戰力**不小於**敵方戰力，他便能打倒對方，使自己的戰力成長為加上對方戰力後的數值。反之，玩家會戰敗，遊戲也會結束。

除了選擇敵人戰鬥外，如果塔上有還沒拾取的寶劍的話，玩家也可以選擇拾取一把寶劍，並讓自己的戰力**成長為當下戰力的兩倍**。當玩家挑戰完塔上的所有敵人，便可依序前往下一座塔。請注意所有寶劍只能在前往下一座塔前被拾取，玩家不可前往前一座塔拾取該塔內尚未拾取的寶劍。

現在，請你幫忙計算一下，如果使用最佳策略遊玩的話，最多可以挑戰完幾座敵方的塔呢？

Input

輸入第一行包含兩個正整數 N 和 X ，代表塔的數量和玩家的初始戰力。

接下來包含 N 行，第 i 行首先會有一個正整數 k_i 和一個非負整數 s_i ，代表第 i 座塔的敵人數量和寶劍數量。接著會有 k_i 個正整數 $a_{i,1}, a_{i,2}, \dots, a_{i,k_i}$ ，代表每個敵人的戰力。

- $1 \leq N \leq 10^5$
- $1 \leq \sum_i (k_i + s_i) \leq 10^6$
- $k_i \geq 1, s_i \geq 0$
- $1 \leq a_{i,j}, X \leq 10^9$

Output

輸出一個非負整數，代表最多可以挑戰完成的塔數量

Sample Input 1

1 22	1
3 0 21 75 38	

Sample Output 1**Sample Input 2**

3 10	3
2 0 5 12	
3 1 48 25 99	
2 0 300 500	

Sample Output 2**Sample Input 3**

3 10	2
2 0 5 12	
3 1 48 25 99	
2 0 300 1000	

Sample Output 3

G. 模仿大師

Problem ID: imitator

Alice 是個著名的模仿大師，她擅長在各種方面對他人進行維妙維肖的模仿。她高超的模仿技術源自於她平時對於自身模仿技術的磨練，也因此，她非常注重模仿技術的練習。

這天，Alice 又為自己找了一個練習的任務，她發現 Bob 手上有一個長度為 N 的數列，並且在數列中，1 到 N 各出現了一次。Alice 決定要試著模仿 Bob 的數列，於是她也構造一個長度為 N ，1 到 N 各出現了一次的數列。然而，她卻發現這個數列跟 Bob 手上的數列長的非常不像。為了不愧對自己模仿大師的稱號，她決定對她目前的數列進行若干次「交換兩相鄰元素」的操作。舉例來說。Alice 可以對 $\{1, 3, 2, 4, 5\}$ 中相鄰的 $(3, 2)$ 做交換操作得到 $\{1, 2, 3, 4, 5\}$ 。

請問你能幫 Alice 計算，她最少需要進行幾次交換操作才能讓自己的數列變成 Bob 的數列，完成模仿呢？

Input

輸入第一行包含一個正整數 N 。

第二行包含 N 個正整數 a_1, a_2, \dots, a_N 代表 Alice 的數列。

第三行包含 N 個正整數 b_1, b_2, \dots, b_N 代表 Bob 的數列。

- $1 \leq N \leq 5000$
- $1 \leq a_i, b_i \leq N$
- $a_i \neq a_j, b_i \neq b_j, \forall i \neq j$

Output

輸出一個整數代表 Alice 完成模仿最少需要的交換次數。

Sample Input 1

Sample Input 1	Sample Output 1
5 2 4 1 3 5 1 2 5 3 4	5

This page is intentionally left blank.

H. 跑操場

Problem ID: laps

在台大的運動場中，有一個長度為 L 公尺的環形跑道。Alice 和 Bob 是兩位正在跑道上跑步的好朋友。



圖：台大運動場的衛星圖。來源：Google Earth。

雖然 Alice 和 Bob 都是順時針方向奔跑，但 Alice 的跑步速度是每秒 v_A 公尺，而 Bob 的速度是每秒 v_B 公尺。此時，Alice 距離起點 x_A 公尺，而 Bob 距離起點 x_B 公尺。雖然他們是好朋友，但雙方都不願改變自己的速度。不過，他們仍然想知道第一次相遇時會在哪裡，因此請你幫他們計算！

如你所見，這是一個相當簡單的問題，可見他們的數學可能不太好，也不懂浮點數或分數。如果他們第一次相遇的位置不是整數公尺，請回報他們相遇之後將跑到的第一個整數公尺位置作為答案。

Input

輸入只有一行，包含五個整數： L 、 v_A 、 v_B 、 x_A 和 x_B 。

- $1 \leq L \leq 10^9$ (是，跑道可以非常長。)
- $1 \leq v_A, v_B \leq 10^9$ (是，他們可以跑得非常快。)
- $0 \leq x_A, x_B < L$
- $x_A \neq x_B$

Output

如果 Alice 和 Bob 永遠不會相遇，輸出 : (。否則，輸出一個整數，表示他們第一次相遇的位置，或是他們相遇後經過的第一個整數公尺位置。

Sample Input 1

10 4 2 3 1

Sample Output 1

9

Sample Input 2

10 4 2 0 1

Sample Output 2

2

Sample Input 3

10 4 2 0 2

Sample Output 3

4

Sample Input 4

10 5 1 0 1

Sample Output 4

2

Sample Input 5

10 1000000000 1000000000 0 9

Sample Output 5

:(

I. 圈圈叉叉

Problem ID: tictactoe

相信大家都很熟悉圈圈叉叉這款經典的紙筆遊戲。不過，在這道題目中，我們將要在一個巨大的網格上玩這個遊戲！

(1,1)	(1,2)	(1,3)	...	(1,N)
(2,1)	(2,2)	(2,3)	...	(2,N)
(3,1)	(3,2)	(3,3)	...	(3,N)

⋮ ⋮ ⋮ ⋮ ⋮

(N,1)	(N,2)	(N,3)	...	(N,N)
-------	-------	-------	-----	-------

相較於一般的版本，我們將會在一個 $N \times N$ 的網格上進行遊戲。兩位玩家和一般的版本一樣輪流在棋盤上劃記，第一位玩家劃記 X，第二位玩家劃記 0。另一個與一般遊戲的差異在於玩家可以在已經劃過的格子上劃記，一個已經被劃記過的格子如果再被劃一次，那新的劃記會覆蓋過原本的。每一步後，每位玩家當前的分數會是完全由他們的劃記所組成的水平、垂直和對角線數量。

現在，給你每一步玩家劃記了哪個座標，請你計算每一步後兩位玩家的分數。

Input

輸入的第一行有兩個正整數 N, Q 。

接下來的 Q 行，每行有兩個正整數 x_i, y_i 。 (x_i, y_i) 表示第 i 步玩家劃記的座標。如果 i 是奇數，則這一步是第一位玩家下的；如果 i 是偶數，則這一步是第二位玩家下的。

- $2 \leq N \leq 1000$

- $1 \leq Q \leq 10^6$
- $1 \leq x_i, y_i \leq N$

Output

輸出 Q 行，每行兩個整數，分別表示在第 i 步後第一位玩家和第二位玩家的分數。

Sample Input 1

3 6
2 2
1 1
1 2
2 1
3 2
3 2

Sample Output 1

0 0
0 0
0 0
0 0
1 0
0 0

Sample Input 2

3 8
1 1
1 3
3 3
3 1
2 2
2 2
2 2
2 2

Sample Output 2

0 0
0 0
0 0
0 0
1 0
0 1
1 0
0 1

J. 牛肉麵店的隊伍

Problem ID: restaurant

臺灣一向以種種美食聞名。不過，就算有那麼多美食，臺灣人還是很喜歡排在長長的隊伍中等著吃飯。

在這家有名的牛肉麵店，廚房一次只能煮一位客人的餐點。第 i 個客人在 t_i 時刻到達，而他點的餐點需要花 w_i 分鐘來煮。當客人到達時，如果廚房沒有正在處理其他客人的餐點，那就會立刻開始煮他的。否則，他將會排到隊伍的最後面。每個人都會好好遵守規則，不會插隊。廚房在前一個餐點煮完的瞬間，馬上就可以開始煮下一個客人的餐點。現在，請計算每位客人各需要等多久才能等到他們的餐點。

Input

第一行有一個正整數 N 。

接下來的 N 行，每行有兩個正整數 t_i, w_i 。

- $1 \leq N \leq 10^5$
- $1 \leq t_i, w_i \leq 10^9$
- t 是嚴格遞增的，換句話說 $\forall 1 \leq i < N, t_i < t_{i+1}$ 。

Output

輸出 N 行，每行一個正整數。第 i 行表示第 i 個客人等了多久才拿到他們的餐點。

Sample Input 1

3
1 5
2 1
7 5

Sample Output 1

5
5
5

This page is intentionally left blank.

K. Git 分支圖

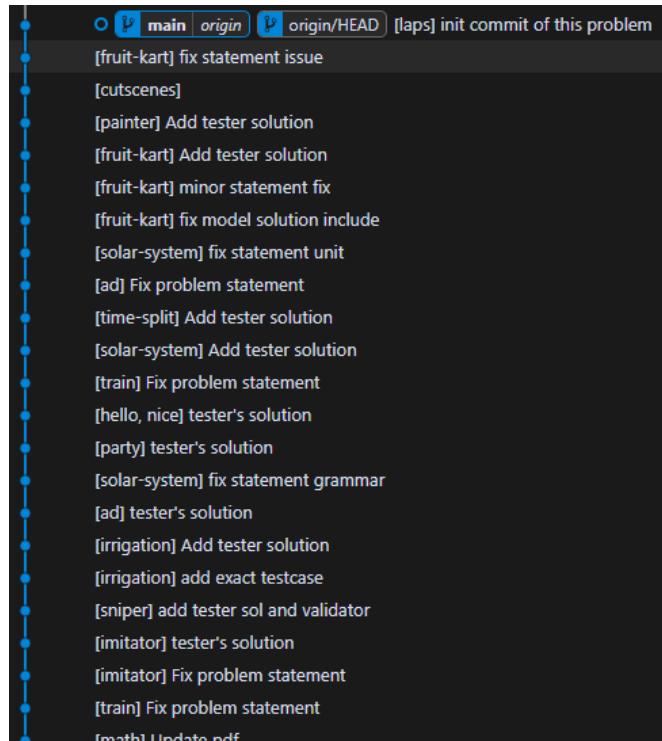
Problem ID: graph

Git 是一個在軟體開發產業中廣泛使用的現代版本控制系統。Git 的一個關鍵功能是建立分支的能力，分支是可以合併回去的獨立開發線。這可以簡化開發過程，使得功能的平行開發可以在不互相干擾的情況下進行。



圖：一個儲存庫中前幾個提交的 Git 圖。

在 Git 儲存庫中，為了編輯程式碼庫，開發者建立提交 (commit) 來記錄他們所做的更改。提交是程式碼庫在某個時間點的快照。更重要的是，除了初始提交之外的所有提交還會記錄父提交的 ID。如果儲存庫的歷史是簡單且線性的，這會形成一條鏈。



圖：一個簡單的 Git 圖。它是線性的，因為沒有建立分支。

然而，在更複雜的儲存庫中，不同的開發者在不同的功能上工作可能會編輯相同的檔案。在這種情況下，他們應該建立分支以避免他們的編輯互相干擾。當一筆新提交的父提交已經有後代的話，就會自動建立分支。這也會導致一個提交成為多個提交的父提交。

在有分支的儲存庫中，開發者應該只在將分支合併回去時一次處理衝突。這會建立一種稱為合併提交的特殊類型提交。合併提交會有兩個父提交。



圖：一個更複雜的 Git 圖。它不是線性的，因為建立了分支並合併回去。

如你所見，要了解儲存庫的歷史，繪製圖表是視覺化歷史的好方法。為了區分不同分支的歷史，我們可以用不同的顏色為不同分支的節點著色。我們可以說，如果一個分支已經合併到另一個分支，則該分支是 死亡的。另外，在圖的右下角（紅色的線）也可以發現，如果一個提交沒有其他提交的父提交是它，則該分支可以被認為在該提交後 死亡。

在分支的整個生命週期中，它應該被分配一個與所有活著的分支的顏色都不同的顏色。每當一個分支 死去，原本該分支被分配的顏色可以再重新被分配給這之後新建立的分支。

現在，我們有了 Git 儲存庫的歷史。然而，我們只有提交的列表。對於每個提交，你會得到時間戳和父提交的索引。你的任務是確定需要多少種顏色來根據所述規則為圖表著色。

Input

輸入的第一行包含一個整數 N ，表示儲存庫中的提交數量。

第二行包含一個整數 T_1 ，表示初始提交的時間戳。

接下來的 $N - 1$ 行輸入，每行代表一個提交。每行以一個字元 $type_i$ 開始。如果 $type_i$ 是 C，後面會跟著兩個整數 T_i 和 P_i 。 T_i 表示第 i 個提交的時間戳， P_i 表示父提交是第 P_i 個提交。

如果 $type_i$ 是 M，後面會跟著三個整數 T_i P_i P'_i 。 T_i 表示第 i 個提交的時間戳， P_i 表示父提交是第 P_i 個提交。 P'_i 表示另一個父提交是第 P'_i 個提交。

- $1 \leq N \leq 10^6$
- $0 \leq T_i \leq 10^9$
- 保證所有的 T_i 都是不同的。
- $1 \leq P_i, P'_i \leq N$
- $P_i \neq i$
- $P'_i \neq i$
- $P_i \neq P'_i$
- $T_{P_i} < T_i$
- $T_{P'_i} < T_i$

Output

輸出包含一個整數，表示最少需要幾種相異的顏色才能為這張歷史圖上色。

Sample Input 1

3 1 C 2 1 M 3 1 2	
----------------------------	--

Sample Output 1

2

Sample Input 2

```
5
1
C 10 1
C 9 1
C 8 1
C 7 1
```

Sample Output 2

```
4
```

Sample Input 3

```
3
1
M 3 1 3
C 2 1
```

Sample Output 3

```
2
```

L. 中樞灌溉系統

Problem ID: irrigation

在設計農業的灌溉設施時，除了傳統的溝渠以外，中樞灌溉系統是一種有效的替代方案，這個方法使用設置在中心的灑水器保持周圍的土壤溼潤，從上方看起來就像一個個的綠色的巨大圓圈，稱為作物圈。



圖：美國堪薩斯州由中樞灌溉系統形成的圓形農場的衛星影像。

水果王國在他們的土地上有 n 塊這樣的作物圈，第 i 個的圓心座落在 (x_i, y_i) 的位置而半徑為 r_i 。這些作物圈可能會相交或甚至互相包含。

傳奇芒果國王是水果王國的國王，為了改善王國的交通系統，他想要在土地上經過 (s_x, s_y) 與 (t_x, t_y) 的直線上興建一條條鐵路，雖然確切的鐵路線段還沒有被決定，國王害怕這條鐵路會傷害王國內傳統農業的發展，因此，國王想要知道有多少作物圈會落在這條直線上面。

由於你是這個國度裡最受人景仰的學者，傳奇芒果國王請求你幫忙完成這項重責大任。

Input

輸入的第一行有五個以空白分開的整數 n, s_x, s_y, t_x, t_y 。接下來的 n 行中每行都有三個以空白分開的整數，第 i 行的整數依序為 x_i, y_i, r_i ，意義如題目所述。

- $1 \leq n \leq 10^5$
- $-10^4 \leq s_x, s_y, t_x, t_y, x_i, y_i \leq 10^4$
- $1 \leq r_i \leq 10^4$
- $(s_x, s_y) \neq (t_x, t_y)$

Output

輸出一個非負整數代表與直線相交的作物圈數量。如果有多個圓心與半徑皆相同的作物圈，相交的數量仍要重複計算。

Sample Input 1

3 0 0 3 4	2
3 -4 5	
-4 -4 1	
0 5 2	

Sample Output 1**Sample Input 2**

3 0 0 0 1	3
0 3 1	
0 3 1	
0 3 1	

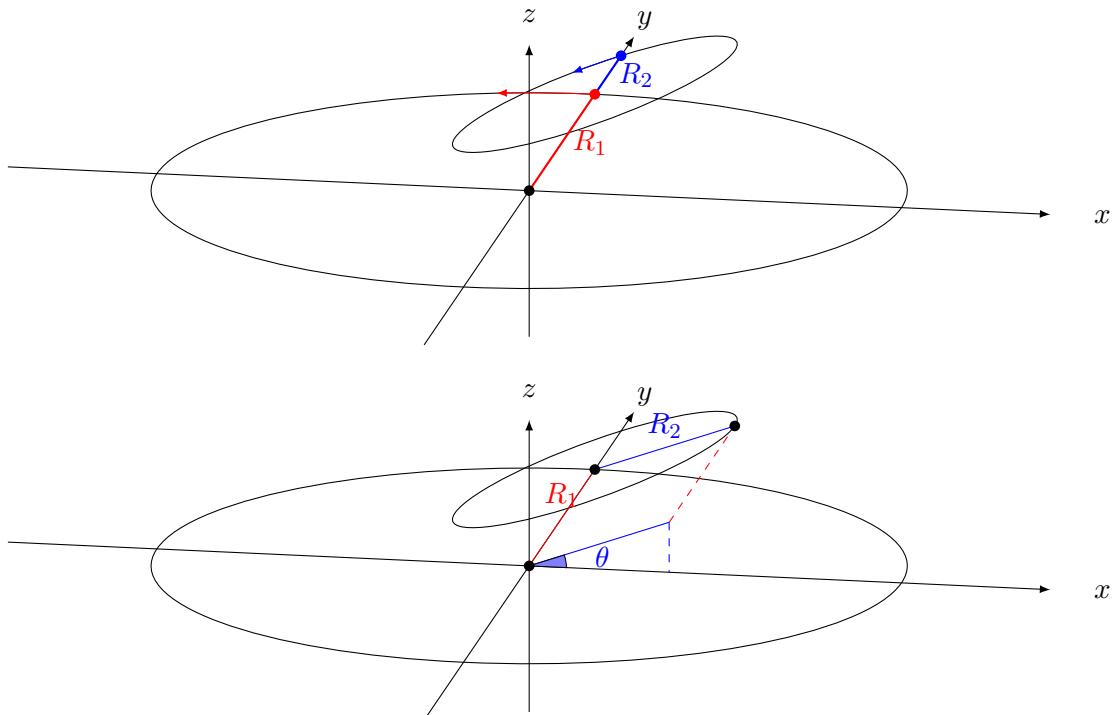
Sample Output 2

M. 太陽系

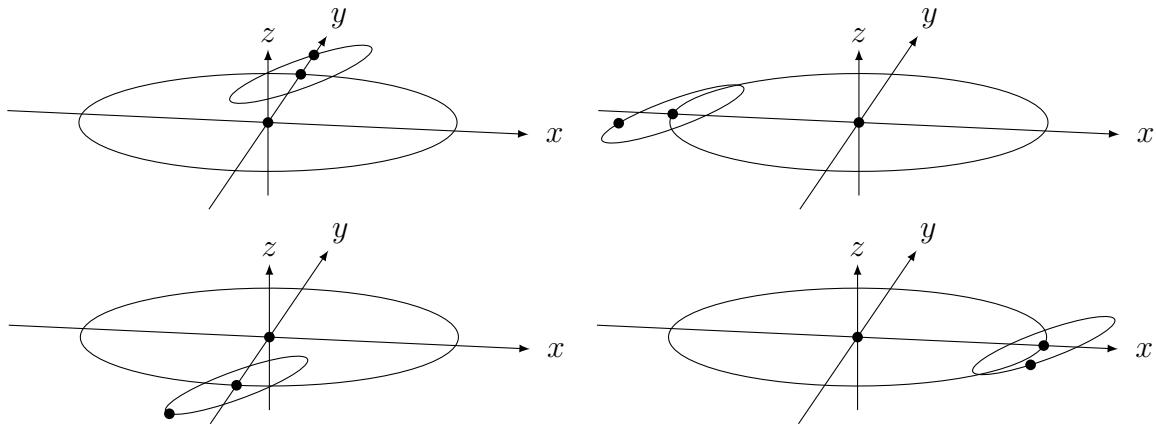
Problem ID: solar-system

在我們的太陽系當中，太陽、地球、以及月亮，形成了一個精巧的系統，在歷史上塑造了我們對於年、月、日的概念。這類行星系統在銀河系中相當典型，行星繞著恆星公轉，衛星繞著行星公轉，當恆星、行星、衛星連成一條線的時候，日食（或月食）就會發生，數以百計的傳說都從這個詭異又迷人的天文事件說起。

以數學的角度來說，我們可以用五個參數近似這樣的系統：兩個軌道的半徑、兩個天體運行的週期、以及兩個軌道平面的夾角。考慮靜止的三維座標，並且讓恆星做為原點而行星在 x - y 平面上半徑為 R_1 的圓。衛星的軌道是以行星為圓心半徑為 R_2 的圓。不過這兩個軌道不一定在同一個平面上，我們用 θ 表示兩個軌道平面的夾角。當行星在恆星的 $+y$ 方向的時候，衛星的軌道會通過 $(0, R_1 - R_2, 0)$, $(0, R_1 + R_2, 0)$ 以及 $(-R_2 \cos \theta, R_1, -R_2 \sin \theta)$ 三個點，如下圖所示。俯瞰時（視線沿著 $-z$ 軸的方向）行星與衛星皆以逆時針等速率公轉。



行星與衛星的軌道關係是固定的，也就是說兩個軌道的夾角不會因為行星公轉而被旋轉，如下圖所示。



如果行星的公轉週期是 T_1 個地球日，而衛星的公轉週期是 T_2 個地球日，從恆星、行星、衛星按照順序在 $+y$ 軸上對齊的時刻算起，經過 t 個地球日後，恆星、行星與衛星的夾角是多少？你需要回答以行星為頂點的那個角度。

Input

輸入只有一行，包含六個以空白分開的整數 $R_1, R_2, T_1, T_2, \theta, t$ 。軌道夾角 θ 是以度數的單位輸入。

- $1 \leq R_1, R_2 \leq 2 \cdot 10^8$
- $1 \leq T_1, T_2 \leq 1000$
- $R_1 \geq 2R_2$
- $0 \leq \theta < 90$
- $0 \leq t \leq 10^9$

Output

輸出 t 個地球日後恆星、行星與衛星的夾角。輸出的單位是度數，並且落在 0 至 180 的範圍中。

你的答案會被視為正確如果相對或絕對誤差不超過 10^{-4} ，正式的說，如果你的答案是 a 而正確答案是 b ，你的答案會被視為正確若且唯若 $\frac{|a-b|}{\max(b,1)} \leq 10^{-4}$ 。

Sample Input 1

20 10 36 8 45 27

Sample Output 1

60

Sample Input 2

149597871 384399 365 27 5 1000

Sample Output 2

73.02050669239190587433

Sample Input 3

149597871 384399 365 27 5 0

Sample Output 3

180

This page is intentionally left blank.

N. 狙擊對決

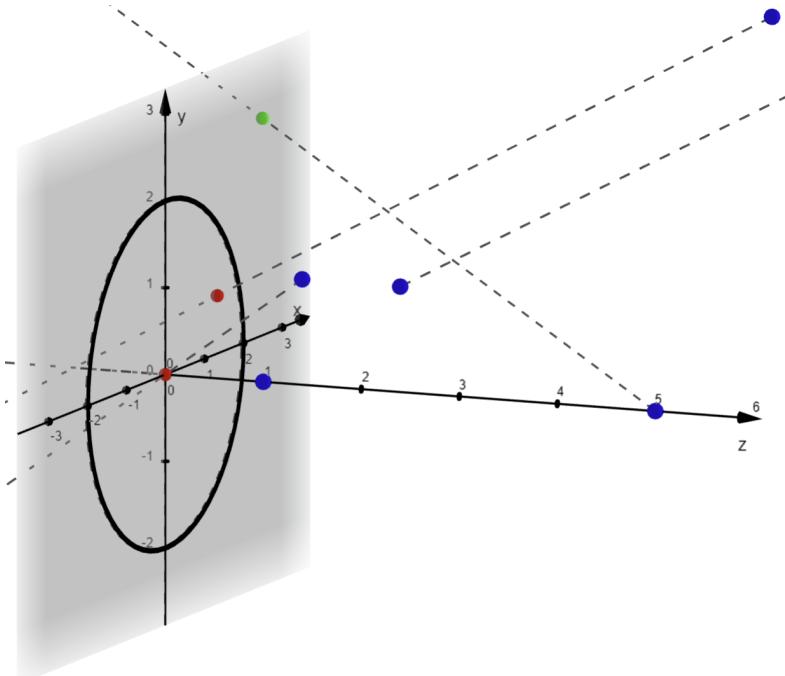
Problem ID: sniper

在虛擬射擊遊戲角戰英雄 (Polygon Warfare) 中， Q 位精英狙擊手們在高科技射擊場進行訓練。該場地中央設有一個位於 $z = 0$ 平面上、中心在原點 $(0, 0, 0)$ 、半徑為 r 的圓形靶心。

這 Q 位狙擊手分別站在靶心前 ($z > 0$) 的某處 (因為是在虛擬遊戲中，他們可以站在同一個點)，並向固定方向射出一發子彈。每顆子彈會沿著一條由其起點和方向向量定義的射線無限延伸。

請判斷每位狙擊手的子彈是否會命中靶心 (即該射線是否與 $z = 0$ 平面上的圓形靶心相交)。

下圖展示了 5 位狙擊手射出的 5 顆子彈，其中藍色點代表子彈的起點，虛線代表子彈的軌跡。



Input

第一行包含兩個以空白分隔的整數 r 和 Q ，分別代表靶心的半徑與狙擊手的數量。接下來的 Q 行，每行包含六個以空白分隔的整數： x, y, z, dx, dy, dz 。其中 (x, y, z) 為子彈的發射起點， (dx, dy, dz) 為子彈的方向向量。

- $1 \leq r \leq 10^9$
- $1 \leq Q \leq 10^5$
- $z > 0$.
- $-10^9 \leq x, y, z, dx, dy, dz \leq 10^9$.
- 保證所有的方向向量 (dx, dy, dz) 都不是零向量。
- 保證沒有子彈的軌跡與 $z = 0$ 平面的交點與原點的距離在 $[(1 - 10^{-6})r, (1 + 10^{-6})r]$ 的範圍內。

Output

對於每位狙擊手，輸出一行”yes”（命中靶心）或”no”（未命中靶心），不含引號。

Sample Input 1	Sample Output 1
2 5 0 0 1 0 0 -1 0 0 5 1 1 -2 1 1 1 -2 -2 -2 1 1 2 2 2 3 3 4 5 -1 -2 -3	yes no yes no yes

O. 股票大亨的獲利追蹤器

Problem ID: stock

傳奇股票大亨戈德斯坦先生 (Mr. Goldstein) 每天都在買賣股票，並細心記錄每天的盈虧。

最近，他希望能快速查詢自己在某一段期間內的總獲利。

請根據他每日的盈虧紀錄，以及多筆查詢。每筆查詢會給出一個區間 l 到 r ，請幫他計算這段期間（含頭尾兩天）的總獲利。

Input

第一行包含一整數 n ，代表天數。第二行包含 n 個整數 p_1, p_2, \dots, p_n ，其中 p_i 代表第 i 天的獲利或虧損。第三行包含一整數 q ，代表查詢的數量。接下來的 q 行，每行包含兩個整數 l 和 r ，表示戈德斯坦先生想要查詢從第 l 天到第 r 天（含頭尾兩天）的總獲利。

- $1 \leq n \leq 10^5$
- $|p_i| \leq 10^9$ for all $1 \leq i \leq n$
- $1 \leq q \leq 10^5$
- $1 \leq l \leq r \leq n$ for all queries

Output

對每筆查詢，輸出一行整數，表示從第 l 天到第 r 天的總獲利。

Sample Input 1	Sample Output 1
5 2 5 18 -7 6 5 1 5 2 4 3 3 1 4 4 5	24 16 18 18 -1

This page is intentionally left blank.

P. 模運算下的聯立方程式

Problem ID: modulo

在初等數學中，我們學過實數的聯立方程式。假設有這樣一個聯立方程式：

$$\left\{ \begin{array}{l} A_{1,1}x_1 + A_{1,2}x_2 + \cdots + A_{1,n}x_n = b_1 \\ A_{2,1}x_1 + A_{2,2}x_2 + \cdots + A_{2,n}x_n = b_2 \\ \vdots \\ A_{n,1}x_1 + A_{n,2}x_2 + \cdots + A_{n,n}x_n = b_n \end{array} \right.$$

這個著名的問題可以用高斯消去法來解決。該方法的概念相當簡單。我們對增廣矩陣 $[A|b]$ 應用基本行運算，即將一行的倍數加到另一行。可以證明，在這個運算之後， $Ax = b$ 的解不會受到影響。然後，為了消去第 n 項，我們查看第 2 到第 n 個方程，對於第 i 個方程，我們找到係數 $c = -\frac{A_{i,n}}{A_{1,n}}$ ，並讓新的第 i 行為：

$$(cA_{1,1} + A_{i,1})x_1 + (cA_{1,2} + A_{i,2})x_2 + \cdots + (cA_{1,n-1} + A_{i,n-1})x_{n-1} + (cA_{1,n} + A_{i,n})x_n = cb_1 + b_i$$

由於 $cA_{1,n} = -A_{i,n}$ ，實際上是：

$$(cA_{1,1} + A_{i,1})x_1 + (cA_{1,2} + A_{i,2})x_2 + \cdots + (cA_{1,n-1} + A_{i,n-1})x_{n-1} + 0x_n = cb_1 + b_i$$

因此第 n 項在第 i 個方程中被消去了。

類似地，這個過程可以套用在第 $n - 1$ 項上。利用新產生的第二個等式，可以消去第三個到最後一個等式中的第 $n - 1$ 項。重複進行這個消去 $n - 1$ 次，會得到新的 A' 和 b' ，並且，滿足 $i > j$ 的 $A'i, j$ 都一定會是零。特別地，第 n 個方程式會變成簡單的 $A'n, 1x_1 = b'_n$ 。在已知 x_1 之後，第 $(n - 1)$ 個方程式也只會剩下一個未知數 x_2 。接著，再從第 n 個方程式開始，回頭依序代入每一個等式到第一個為止，即可依序解出所有 x 。整個過程僅需 $O(n^3)$ 次運算即可完成。

這個方法直接且易於理解。然而，由於使用了除法，我們實際上無法在模運算的世界中使用這個方法來求解。但，我們真的不能嗎？

事實上，雖然除法通常在模運算的世界中不能直覺的使用，但我們可以用某種方式來定義除法。當我們思考除法的定義 $\frac{a}{b} = x$ 時，它實際上是找到 x 使得 $x \times b = a$ 。有了這個想法，在模運算的世界中，我們也可以嘗試找到 x 使得 $x \times b \equiv a \pmod{m}$ 。為了做到這一點，我們必須利用費馬小定理。

費馬小定理指出，對於質數 p 和任何整數 a , $(1 \leq a < p)$, $a^p \equiv a \pmod{p}$ 。有了這個，我們也知道 $a^{p-1} \equiv 1 \pmod{p}$ 。現在，為了找到 x 使得 $x \times b \equiv a \pmod{p}$ ，我們可以證明 $x = a \times b^{p-2} \pmod{p}$ 是一個有效的解。

剩下要解決的最後一個問題是如何計算 $b^{p-2} \pmod{p}$ 。天真地進行計算需要 $O(p)$ 時間，這太慢了（在這個問題中， m 可以大到 10^9 ！）。不過這可以用快速幕的技巧來完成。

這個技巧適用於任何帶有整數指數的一般指數計算。為了計算 a^b ，我們可以計算 $a^0, a^1, a^2, a^4, a^8, \dots, a^{2^k}$ ，其中 $k = \lfloor \log_2 b \rfloor$ 。然後，由於我們可以將 b 分解為 2 的幕次和，我們可以通過乘以適當的 a 的幕來計算 a^b 。這可以在 $O(\log b)$ 時間內完成。將模運算應用於這個技巧不會改變複雜度。因此，我們可以在 $O(n^3 \log m)$ 時間內解決聯立方程式問題。現在，你能解決它嗎？

Input

輸入的第一行包含兩個整數 n, m 。

接下來的 n 行描述方程式。每行包含 $n + 1$ 個整數：前 n 個整數是第 i 個方程的係數 $A_{i,1}, A_{i,2}, \dots, A_{i,n}$ ，最後一個整數是 b_i 。每行代表一個方程 $A_{i,1}x_1 + A_{i,2}x_2 + \dots + A_{i,n}x_n = b_i$ 。

- $1 \leq n \leq 100$
- $2 \leq m \leq 10^9$ ， m 是一個質數。
- $\forall i, j, 0 \leq A_{i,j} < m$
- $\forall i, 0 \leq b_i < m$

Output

如果該系統有唯一解，在一行上輸出 n 個整數，表示 x_1, x_2, \dots, x_n 模 m 的值。

如果該系統無解，輸出 NO SOLUTION。

如果該系統有多於一個解，輸出 MORE THAN ONE SOLUTION。

Sample Input 1

3 998244353
1 1 0 2
0 1 0 1
0 1 2 2

Sample Output 1

1 1 499122177

Sample Input 2

2 7	6 2
1 2 3	
4 5 6	

Sample Output 2**Sample Input 3**

2 5	NO SOLUTION
1 2 3	
2 4 2	

Sample Output 3**Sample Input 4**

3 7	MORE THAN ONE SOLUTION
1 2 0 3	
2 4 0 6	
0 0 1 4	

Sample Output 4**Sample Input 5**

1 7	4
3 5	

Sample Output 5

This page is intentionally left blank.

Q. 火車運輸系統

Problem ID: train

「火花奇遇記：自動化冒險」是一款開放世界工藝生存類型的遊戲。在遊戲中，你可以收集各種資源，發揮創造力設計並建造自動化工坊，享受自動化帶來的樂趣。



小奕在遊玩遊戲時發現，為了讓工坊能達成自動化，他需要某種資源 A 源源不絕的出現在位置 Y ，然而，資源 A 只會在非常遙遠的位置 X 無限產生。為了解決這個問題，小奕決定鋪設環形鐵路，來讓火車可以將資源 A 從位置 X 運輸至位置 Y ，再返回到位置 X 重複運輸。

不過，新的問題隨之出現了，小奕很快發現到，除了資源 A 以外，他還需要一種只會在位置 X' 無限產生的資源 B ，源源不絕的出現在位置 Y' 。由於物資有限，小奕只足夠設置一台火車和一組環形軌道，火車一次只能夠裝載一種資源，請問你可以幫忙設計環形軌道來讓兩種資源都能夠被完成運輸嗎？

正式地說，在一個 $N \times M$ 的表格上，資源 A 要從 (x_{a_1}, y_{a_1}) 運輸到 (x_{a_2}, y_{a_2}) ，資源 B 要從 (x_{b_1}, y_{b_1}) 運輸到 (x_{b_2}, y_{b_2}) ，一組滿足要求的環形軌道 $R = \{(x_1, y_1), (x_2, y_2), \dots, (x_{|R|}, y_{|R|})\}$ 需滿足：

- $1 \leq x_i \leq N, 1 \leq y_i \leq M, 1 \leq i \leq |R|$
- $\forall 1 \leq i, j \leq |R|, i \neq j, (x_i, y_i) \neq (x_j, y_j)$
- (x_i, y_i) 與 (x_{i+1}, y_{i+1}) 相鄰， $1 \leq i < |R|$
- $(x_{|R|}, y_{|R|})$ 與 (x_1, y_1) 相鄰
- $\{(x_{a_1}, y_{a_1}), (x_{a_2}, y_{a_2}), (x_{b_1}, y_{b_1}), (x_{b_2}, y_{b_2})\} \subseteq R$

注意到，這裡的相鄰指的是四方位相鄰，及 (x_i, y_i) 和 (x_j, y_j) 相鄰若且唯若 $|x_i - x_j| + |y_i - y_j| = 1$ 。

並且若是從 (x_{a_1}, y_{a_1}) 開始沿著環形軌道行進，必定接著依序經過 $(x_{a_2}, y_{a_2}), (x_{b_1}, y_{b_1}), (x_{b_2}, y_{b_2})$ 。保證給定的表格大小及位置必定能找到至少一組滿足要求的環形軌道，如果有多種滿足要求的環狀路線，輸出任意一種即可。

Input

輸入第一行包含一行兩個正整數 N 和 M 代表表格的大小。

接下來有四行輸入，每一行有兩個正整數，分別代表 $(x_{a_1}, y_{a_1}), (x_{a_2}, y_{a_2}), (x_{b_1}, y_{b_1}), (x_{b_2}, y_{b_2})$ 。並且，保證這四行代表的座標皆不相同

- $2 \leq N, M \leq 5$
- $1 \leq x_{a_1}, x_{a_2}, x_{b_1}, x_{b_2} \leq N$
- $1 \leq y_{a_1}, y_{a_2}, y_{b_1}, y_{b_2} \leq M$

Output

輸出第一行為一個正整數 $|R|$ 代表環狀軌道 R 的長度。

接下來輸出 $|R|$ 行，第 i 行為兩個正整數 x_i, y_i ，代表環狀軌道 R 中的 (x_i, y_i) 。

Sample Input 1

Sample Input 1	Sample Output 1
<pre>3 3 1 1 1 3 2 3 3 1</pre>	<pre>8 1 1 1 2 1 3 2 3 3 3 3 2 3 1 2 1</pre>

Sample Input 2

```
5 5
2 2
4 4
2 4
4 2
```

Sample Output 2

```
20
2 2
1 2
1 1
2 1
3 1
4 1
5 1
5 2
5 3
5 4
4 4
3 4
2 4
1 4
1 3
2 3
3 3
4 3
4 2
3 2
```

This page is intentionally left blank.

R. 機密竊取

Problem ID: intel

作為一名被派遣至敵軍後方的菁英特種部隊成員，你的任務是潛入敵方領地、取得關鍵情報並安全撤離：

- 從基地 (S) 出發，
- 抵達情報點 (T) 並取得機密資料，
- 最後前往撤離點 (E) 完成撤離。

戰場由牆壁與通路組成的迷宮構成。你每次可以向上、下、左、右移動一步，但不能斜向移動，也不能穿越牆壁。

請計算從基地 (S) 出發，經過情報點 (T)，最終到達撤離點 (E) 所需的最少步數。若不可能完成任務，回報無法達成。

Input

第一行包含兩個以空格分隔的整數 n 和 m ，代表迷宮的行數與列數。接下來的 n 行中，每行一個長度為 m 的字串，代表迷宮的每一行，其中：

- # 代表牆壁，
 - . 代表通路，
 - S 代表基地/起點，
 - T 代表情報點，
 - E 代表撤離/出口點。
-
- $4 \leq n, m \leq 1000$ 。
 - 保證迷宮中恰有一個基地 (S)、一個情報點 (T) 和一個撤離點 (E)。
 - 迷宮的最外層邊界全部由牆壁 (#) 組成。

Output

輸出一個整數，表示從基地 (S) 經過情報點 (T) 到達撤離點 (E) 所需的最少步數。若不可能完成任務，請輸出 -1。

Sample Input 1

```
5 7
#####
#S...E#
###.#.#
#T....#
#####
```

Sample Output 1

```
12
```

S. 分段時間

Problem ID: time-split

電動遊戲是二十一世紀中規模最大的娛樂活動之一，而競速破關 (speedrunning，又稱速通) 絕對是時下其中一個體會這些數位遊戲最受歡迎的方法。在競速破關的領域中，紀錄各段過程的時間花費是絕對不可或缺的工具之一，這個工具會紀錄每一段遊戲過程或者是達成某個里程碑的時間，讓玩家得以檢討以及改進當下的紀錄。

Nathan 是一位競速破關社群中的閃耀新星，不僅如此，他還有自己擁有一套紀錄分段時間的工具 Splee，Splee 以 $hh:mm:ss.sss$ 的格式紀錄時間，詳細說明如下：

- hh 是一個二位數字，紀錄花費的小時數。
- mm 是一個二位數字，紀錄花費的分鐘數。
- $ss.sss$ 是一個小數點前有兩位、小數點後有三位的數字，紀錄花費的秒數。
- 需要時補上前導零與後綴零以填滿所有需要的位數。
- 小時數必須介於 0 與 99 之間、分鐘數必須介於 0 之間 59、而秒數必須介於 0 與 59.999 之間。

除此之外，為了方便，這些時間紀錄與顯示上有兩個額外的規則：

- 任意只有數字零與冒號的前綴可以忽略，但是時間不可以以冒號作為開頭。
- 小數點後的後綴零可以忽略，若小數點部份為零，那小數點也可以忽略。但是時間不允許使用單一個小數點或者空字串表示。

舉例來說，一小時又五秒可以以 01:00:05.000、1:00:05.000、或 1:00:05 表示。五分之一秒可以以完整形式表示：00:00:00.200，或者以非常簡潔的形式表示：.2。不過，:00.200 不是一個合法的表示方式。這三個字串 0、.0、或 0. 都可以用來表示完全沒有花費任何時間，但是 . 以及空字串是不被允許的。

在 Nathan 的最近一次世界紀錄嘗試中，其中一段的時間在過程中被弄丟了，幸好他還有總共花費的時間以及其他所有分段的時間，以 Splee 的格式紀錄，你可以設計一隻程式用這些資訊回復弄丟的那段時間嗎？

Input

輸入的第一行只有一個整數 n 代表總共有 n 個段落。在接下來的 n 行中，每一行可能是一個合法的時間標記，代表該分段的時間，或單一個問號 (?)，代表這一段的時間被弄丟了。

在這之後還有一行，含有一個合法的時間標記，代表整個世界紀錄嘗試的總時間。

- $1 \leq n \leq 100$
- 恰好只有一行是單一一個問號 (?)。
- 所有已知的分段時間總和不超過整個世界紀錄嘗試的總時間。

Output

輸出一個字串於一行，以題目中時間標記的格式代表被弄丟的分段時間。如果有多種表示方法，任意一種皆會被視為正確。

Sample Input 1

```
7
1:39
0:23
1:56
0:02
0:44
0:58
?
6:50
```

Sample Output 1

```
1:08
```

Sample Input 2

```
6
?
10
.100
00.010
0:01:00.
00:00:00.001
1:11.111
```

Sample Output 2

```
00:00:01.000
```

T. 外送系統優化

Problem ID: deliver

你正在為一間提供無人機遞送服務的公司設計排程系統。每位顧客都有一組無人機起飛時間，表示無人機可以出發為他們遞送貨物的時間點。

為了提升無人機的整體調度效率，你需要找出一段時間區間 $[l, r]$ ，使得每位顧客至少有一個起飛時間落在此區間內。

請找出滿足此條件的最短區間 $[l, r]$ 。若有多個區間長度相同，請選擇 l 最小的區間。

Input

第一行輸入一個整數 k ，代表顧客的數量。接下來 k 行中的第 i 行包含一個整數 n_i ，代表第 i 位顧客的起飛時間數量，後面跟著 n_i 個相異整數，表示這些起飛時間。

- $1 \leq k \leq 10^5$
- $1 \leq n_i \leq 10^5$
- $\sum_{i=1}^k n_i \leq 10^5$
- 每個起飛時間都是不超過 10^9 的正整數。

Output

請輸出兩個整數 l 與 r ，表示一個最短的時間區間 $[l, r]$ ，使得每位顧客至少有一個起飛時間位於該區間內。

Sample Input 1

3	4 5
5 4 10 24 26 15	4 5
4 4 9 12 20	4 5
5 5 22 30 18 9	4 5

Sample Output 1

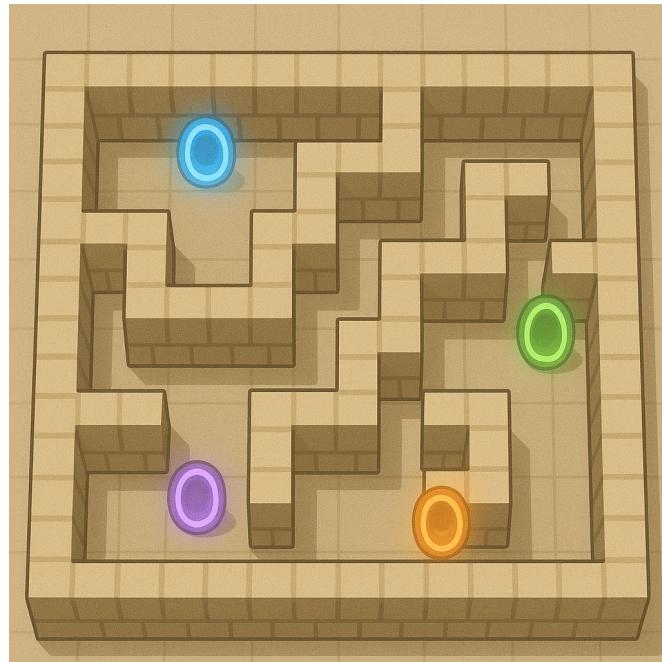
3	4 5
5 4 10 24 26 15	4 5
4 4 9 12 20	4 5
5 5 22 30 18 9	4 5

This page is intentionally left blank.

U. 傳送門

Problem ID: portal

你被困在一個大小為 $N \times M$ 的網格狀迷宮中。



圖：一個包含一些傳送門的迷宮。

在這個迷宮中，每個格子有可能是空的、牆壁、或是一個傳送門。你的任務是從左上角（格子 $(1, 1)$ ）開始，走到右下角（格子 (N, M) ）。你每一步可以朝右邊或下方移動到相鄰的非牆壁的格子。當你進入一個包含傳送門的格子時，你可以選擇進入傳送門，並被傳送到任何位於更右邊和更下方（或同一行或列）且不是牆壁的格子。更精確地說，如果你在有傳送門的格子 (x, y) 上，你可以選擇正常行走，或是傳送到另一個使 $x' \geq x$ 且 $y' \geq y$ 的非牆壁格子 (x', y') 。

另外，有 K 個寶箱在迷宮的一些非牆壁格子中。當你在一個有寶箱的格子時，你會打開寶箱並獲得寶箱的價值。然而，寶箱可能是被詛咒的，所以你有可能從中獲得負的價值。當你進入一個有寶箱的格子時，即使你不想要，你也必須打開寶箱。

Input

輸入的第一行包含三個整數 N 、 M 和 K 。

接下來的 N 行每行包含一個 M 個字元的字串。第 i 行的第 j 個字元表示位置 (i, j) 的格子。字元要麼是 .、x 或 p。

接下來的 K 行每行包含三個整數 x_i 、 y_i 和 v_i 。它表示在位置 (x_i, y_i) 的格子有一個價值為 v_i 的寶箱。

- $1 \leq N, M \leq 1000$
- $0 \leq K \leq N \times M$
- $1 \leq x_i \leq N$
- $1 \leq y_i \leq M$
- $-10^9 \leq v_i \leq 10^9$
- 保證所有 (x_i, y_i) 都是不同的且不是牆壁。
- 保證 $(1, 1)$ 和 (N, M) 不是牆壁。
- 保證存在至少一條從 $(1, 1)$ 到 (N, M) 的路徑。

Output

輸出一個整數，即你能獲得的最大價值。

Sample Input 1

```
3 3 3
...
.x.
...
1 3 2
3 1 -5
3 2 8
```

Sample Output 1

```
3
```

Sample Input 2

```
1 4 2
.pp.
1 3 -1
1 4 2
```

Sample Output 2

```
2
```

V. 數學之神

Problem ID: math

Fysty 是古希臘神話中掌管質因數的神，他控制著宇宙萬物間所有數字比次之間的關係，讓所有的數字都維持他們應有的樣子。

今天，宇宙間的數字突然失去了平衡，導致世界開始動盪不平，為了避免事態持續惡化，Fysty 需要動用他的神力，來把造成一切不平行的數字 X 變成另一個特定的數字 Y 。

Fysty 具體使用神力的方法有兩種，他可以把一個數字 x 乘上某個質因數 a ，並根據該數字 x 擁有的質因數 a 幂次 b 消耗 $b + 1$ 數學之力。舉例而言，如果想把 $2^6 = 64$ 乘上 2 需要花費 7 數學之力。他也可以把一個數字 x 除以某個自己的質因數 a ，並根據該數字 x 擁有的質因數 a 幂次 b 消耗 b 數學之力。舉例而言，如果想把 $2^7 = 128$ 除以 2 需要花費 7 數學之力。

並且，由於 42 是宇宙萬物唯一的答案，Fysty 不能把數字變成任何小於 42 的答案。同時，為維護宇宙間數字的和諧，整個過程中，他都不能讓數字比原先的 X 還要大。

請問你能幫 Fysty 算出，完成這個任務最少需要花費多少數學之力嗎？特別的，如果這個任務無法達成，輸出 -1 。

Input

輸入僅包含一行兩個整數 X 和 Y 。

- $42 \leq Y \leq X \leq 2 \times 10^5$

Output

輸出一個整數代表完成這個任務最少需要花費的數學之力。如果這個任務無法達成，輸出 -1 。

Sample Input 1

864 180	16
---------	----

Sample Output 1

Sample Input 2

1000 500

Sample Output 2

3

W. 水果賽車

Problem ID: fruit-kart

水果賽車是一款刺激的賽車遊戲，支援數千人同時線上遊玩，不過這款遊戲只支援單人競賽，對於想要組隊享受這款遊戲的玩家來說相當沮喪。為了解決這個問題，水果賽車競技社群定好了一套用來組隊參賽的規則：

- 所有隊伍的遊戲名稱要有同一個隊伍前綴。
- 同一隊的所有隊員的最長遊戲名稱共同前綴稱為**隊伍標記**。
- 任意兩隊的隊伍標記不可以以相同的字元開始。

儘管這個方法非常直覺，計算分數以及統計隊伍排名並不是件簡單的事情，為了支援線上賽容易遇到的狀況，例如斷線或者犯規等等，這個社群決定了以下用來紀錄比賽與排名的格式：

- 紀錄中的每一個行由一位玩家的遊戲名稱與分數紀錄組成。每位玩家恰好有一行紀錄。
- 分數紀錄是一格表達式，由數字、加號與減號組成。加減號不會連續出現。
 - 當一個數字前面沒有符號或是一個加號，這個數字是原始得分的一部分。一位玩家的原始得分是所有這類數字的總和。
 - 當一個數字前面是一個減號，這個數字是懲罰分數的一部分。一位玩家的懲罰分數是所有這類負數的總和。
- 一位玩家的總分是他的原始得分加上他的懲罰分數。
- 一支隊伍的總分是所有隊員的總分的總和。
- 玩家與隊伍都以分數排名，高分者排名較高，平手時所有玩家或隊伍都會得到相同排名並跳過中間的排名。

儘管有這樣的排名規則，有時候還是難以一眼看出誰是獲勝的隊伍，這就是為什麼需要表格製作工具：表格製作工具會計算好所有分數與隊伍標記，並且以七個欄位的方式呈現：

- 表格的每一列是一個玩家，這些列會先以隊伍排名排序，再以玩家排名排序。有相同名次時，以先出現在紀錄中的隊伍或玩家出現在前。
- 每一個欄位的寬度恰好是該欄裡最長的格子的長度。
- 每個格子的內容向左對齊，並且加上空格直到該欄位的寬度，除了最後一欄的格子後不額外加上空格。
- 每兩個相鄰的欄位間需要額外插入一個空格字元。

- 第一、第二、第三欄的內容依序是隊伍名次、隊伍標記、以及隊伍總分。這些內容只會在每個隊伍的第一列出現。
- 第四、第五、第六、第七欄的內容依序是玩家名稱、玩家原始得分、玩家懲罰分數、以及玩家排名。
- 第一欄以及最後一欄的名次以 $\#r$ 表示，其中 r 是名次。
- 第六欄中如果該玩家沒有懲罰分數，則留空，否則顯示一對圓括號包住的懲罰分數。

有了這個表格，比賽的各項數據就變得相當顯而易見了，你能幫忙這個社群設計一個強大的表格製作工具嗎？

Input

輸入的第一行有一個整數 n ，代表遊戲紀錄的行數。接下來 n 行中每行有一個紀錄，以題目中敘述的格式呈現。

- $1 \leq n \leq 2000$
- 紀錄中至少有兩個隊伍。
- 玩家名稱中只有 ASCII 編碼介於 33 到 126 中的字元（可顯示字元除去空白字元）。
- 玩家名稱的長度介於 1 與 2000 之間。
- 玩家的原始得分與懲罰分數絕對值不超過 10^9 。
- 每個玩家的分數紀錄長度介於 1 與 2000 之間。
- 輸入的分數紀錄不會有前導零。請注意單一一個零仍舊可以出現。

Output

根據題目敘述的格式輸出表格。

Sample Input 1	Sample Output 1
12 RRLight 74 SS-Markus 97 #always 56 SS-Charlie 106 figure 68 #never 109 ==GOAT== 72 RRDark 94 ==FOUL== 26+20+16-10 discuss 84 figlover 89 disadvantage 83	#1 SS- 203 SS-Charlie 106 #2 SS-Markus 97 #3 #2 RR 168 RRDark 94 #4 RRLight 74 #8 #3 dis 167 discuss 84 #6 disadvantage 83 #7 #4 # 165 #never 109 #1 #always 56 #11 #5 fig 157 figlover 89 #5 figure 68 #10 #6 == 124 ==GOAT== 72 #9 ==FOUL== 62 (-10) #12

Sample Input 2	Sample Output 2
12 (:P) 0+0+0-0-0-0 (OAO) 1+0+0-0+1 average 27 Annoying 33 Astute 33 SoloShow 10+10+12+100 aversion 12 (((.o.))) 15+15+16 Airplane 33 Awful 33 (_._)'' 12+14 averylongname 26+20	#1 A 132 Annoying 33 #4 Astute 33 #4 Airplane 33 #4 Awful 33 #4 #1 SoloShow 132 SoloShow 132 #1 #3 aver 85 averylongname 46 #2 average 27 #8 aversion 12 #10 #4 (74 (((.o.))) 46 #2 (_._)'' 26 #9 (OAO) 2 #11 (:P) 0 #12

Sample Input 3	Sample Output 3
3 A -1 A -1 B -2	#1 A -2 A 0 (-1) #1 A 0 (-1) #1 #1 B -2 B 0 (-2) #3

This page is intentionally left blank.

X. 平衡之森

Problem ID: trees

在神秘的努瑪利亞 (Numaria) 國度裡，精靈森林議會正計劃著恢復魔法森林的平衡。在一條直線森林小徑 (數線) 上，已有 n 棵古老的大樹，分別生長在此線上不同的整數點位置。

然而，精靈們發現部分樹與樹之間的間距過大，導致魔力流動不穩。為了回復和諧，他們希望在現有的樹木之間（不能在第一顆之前或最後一顆之後）種下最多 k 棵在整數點上的新樹，使得樹與樹之間的間距更加均勻，也就是最小化在種新的樹後，任意兩棵相鄰樹之間的最大距離。

請幫精靈們計算出在種下最多 k 棵新樹之後，最遠兩棵相鄰的樹之間距離的最小值。

Input

第一行包含兩整數 n, k ，代表現已有的樹木數量和最多可以種下的新樹數量。第二行包含 n 個整數 p_1, p_2, \dots, p_n ，代表現有樹木在直線上的位置。

- $2 \leq n \leq 10^5$
- $0 \leq k \leq 10^9$
- $-10^9 \leq p_1 < p_2 < \dots < p_n \leq 10^9$

Output

輸出一行，包含一個整數，代表在種下最多 k 棵新樹之後，最遠兩棵相鄰的樹之間距離的最小值。

Sample Input 1

2 1	10
-8 12	

Sample Output 1

Sample Input 2

5 9	1
1 2 3 4 6	

Sample Input 3

5 4 -20 -18 -4 6 9	5
-----------------------	---

Sample Output 3

Y. 畫家

Problem ID: painter

小方塊有一張空白的長方形表格，這格表個有 $w \times h$ 個格子，每個格子都是單位邊長的正方形。如果我們把左下角作為二維座標的原點，整張表個會佔據滿足 $0 \leq x \leq w$ 與 $0 \leq y \leq h$ 的區域。我們用格子 (u, v) 代表區域滿足 $u \leq x \leq u + 1$ 與 $v \leq y \leq v + 1$ 的正方形。

小方塊之前只用直線畫了一張傑作，在每一步當中，他從格子 (x_1, y_1) 與格子 (x_2, y_2) 的中心為端點畫了一條線段，每個與這個線段有相交的格子都會被塗上這條線的顏色，如果發生 $(x_1, y_1) = (x_2, y_2)$ 的狀況，這條線會退化成一個點，這時候只有格子 (x_1, y_1) 會被塗上顏色。

然而，小方塊把這幅畫作弄丟了，所幸他還有原本作畫的操作紀錄，你能將他的傑作復原嗎？

Input

輸入的第一行有三個整數 n, w, h ，代表所有作畫的操作、網格的寬度與高度。接下來有 n 行，每一行有四個整數與一個大寫英文字母字元 x_1, y_1, x_2, y_2, c ，代表這次操作中，小方塊畫了一條以格子 (x_1, y_1) 與格子 (x_2, y_2) 的中心為端點的線段，這條線的顏色是 c 。所有的操作是按照順序輸入的。

- $1 \leq n, w, h \leq 5000$
- 對於所有操作 $0 \leq x_1, x_2 < w$ 與 $0 \leq y_1, y_2 < h$ 。

Output

你需要輸出 h 行，每行有 w 個字元（不含結尾的空白字元）第 i 行的第 j 個字元（皆從 1 開始數起）要是格子 $(h - i, j - 1)$ 的顏色。如果某個格子從來都沒有被著色過，對於該格子請輸出一個點（.）作為他的顏色。

Sample Input 1

```
5 10 10
0 0 9 9 A
1 2 3 4 B
1 8 7 1 C
2 8 9 8 D
8 1 8 1 E
```

Sample Output 1

```
.....AA
.CDDDDDDDD
.CC...AAA.
..CC.AAA..
...CCAA...
..BBCC...
.BBBACC...
.BBA..CC..
AAA....CE.
AA.....
```

Z. 過場動畫

Problem ID: cutscenes

你還記得前幾道題目中的水果賽車嗎？現在小方塊正在使用另一種方式享受這個遊戲：全賽道速通！

在這個遊戲中，賽道被安排在一個 $n \times m$ 的表格上，每一個格子是一個賽道，也就是說，對於所有 $1 \leq i \leq n$ 與 $1 \leq j \leq m$ 都有一個賽道 (i, j) 。全賽道速通顧名思義，就是在最少的時間內遊玩每個賽道至少一遍。不過，在這個遊戲中，不僅僅是遊玩賽道會花費時間：小方塊發現在連續遊玩兩個相近的賽道時，遊戲就會播放玩家從一個賽道開到另一個的過場動畫，不僅對速通玩家来说相當無聊，還耗費了大量的時間。因此，在嘗試速通遊戲時，遊玩賽道的順序會對結果產生重大的影響！

除了練習賽車技術以外，小方塊同時也想知道遊戲的各種機制。經過實驗後，他發現這個遊戲的過場動畫有 k 種，每一種過場動畫都有三個參數 $\Delta x_i, \Delta y_i, t_i$ ，這種過場動畫會在從任意賽道 (x, y) 移動到 $(x + \Delta x_i, y + \Delta y_i)$ 的時候播放，無論起始或終點的賽道是什麼，都會花上 t_i 單位的時間。

小方塊想要在每個賽道都遊玩**恰好一次**的前提下最小化觀看過場動畫的時間，因為遊玩賽道還是最耗費時間的部份。你能幫他找到這樣的一條路線嗎？

Input

輸入的第一行有兩個正整數 n, m 代表表格的長寬。輸入的第二行只有一個整數 k ，代表過場動畫的種類數。接下來有 k 行，第 i 行中有三個整數 $\Delta x_i, \Delta y_i, t_i$ 。

- $1 \leq n, m \leq 1000$
- $0 \leq k \leq 24$
- $0 \leq |\Delta x_i|, |\Delta y_i| \leq 2$
- $1 \leq t_i \leq 10^8$
- 所有 $(\Delta x_i, \Delta y_i)$ 都兩兩相異且不是 $(0, 0)$ 。

Output

第一行請輸出一個整數表示所有路線中觀看過場動畫時間的最小值。第二行請依序輸出 $2 \times n \times m$ 個整數 $x_1, y_1, x_2, y_2 \dots, x_{nm}, y_{nm}$ ，代表其中一條達到最小值的路線是按照

$(x_1, y_1), (x_2, y_2), \dots (x_{nm}, y_{nm})$ 的順序遊玩。

Sample Input 1

5 2
1
1 0 10

Sample Output 1

0
5 2 5 1 4 2 4 1 3 2 3 1 2 2 2 1 1 2 1 1

Sample Input 2

2 2
4
0 1 10
1 0 20
-1 0 30
0 -1 40

Sample Output 2

10
1 2 2 1 2 2 1 1

Sample Input 3

4 1
4
-2 0 30
-1 0 50
1 0 70
2 0 40

Sample Output 3

60
3 1 1 1 4 1 2 1

Sample Input 4

2 3
7
1 0 100
-1 -1 10
0 -1 100
1 -1 100
-1 1 100
0 1 100
1 1 100

Sample Output 4

20
2 2 1 1 1 3 2 1 2 3 1 2