

2018 網際網路程式設計全國大賽

高中組初賽

- 本次比賽共 7 題，含本封面共 20 頁。
- 全部題目的輸入都來自**標準輸入**。輸入中可能包含多組輸入，以題目敘述為主。
- 全部題目的輸出皆輸出到螢幕 (**標準輸出**)。
輸出和裁判的答案必須完全一致，英文字母大小寫不同或有多餘字元皆視為答題錯誤。
- 比賽中上傳之程式碼，使用 C 語言請用 `.c` 為副檔名；使用 C++ 語言則用 `.cpp` 為副檔名。
- 使用 `cin` 輸入速度遠慢於 `scanf` 輸入，若使用需自行承擔 Time Limit Exceeded 的風險。
- 部分題目有浮點數輸出，會採容許部分誤差的方式進行評測。一般來說「相對或絕對誤差小於 ϵ 皆視為正確」， ϵ 值以題目敘述為主。
舉例來說，假設 $\epsilon = 10^{-6}$ 且 a 是正確答案， b 是你的答案，如果符合 $\frac{|a-b|}{\max(|a|,|b|,1)} \leq 10^{-6}$ ，就會被評測程式視為正確。

	題目名稱	時間限制 (秒)
題目 A	差不多是樹堆	2
題目 B	反分裂二元樹法	6
題目 C	領地征戰	5
題目 D	分裂吧，樹堆	1
題目 E	三角形	1
題目 F	十三億人都驚呆了！	10
題目 G	爆裂遊戲	2

2018 網際網路程式設計全國大賽

輸入輸出範例

C 程式範例：

```
1 #include <stdio.h>
2 int main()
3 {
4     int cases;
5     scanf("%d", &cases);
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         scanf("%lld %lld", &a, &b);
10        printf("%lld\n", a + b);
11    }
12    return 0;
13 }
```

C++ 程式範例：

```
1 #include <iostream>
2 int main()
3 {
4     int cases;
5     std::cin >> cases;
6     for (int i = 0; i < cases; ++i)
7     {
8         long long a, b;
9         std::cin >> a >> b;
10        std::cout << a + b << std::endl;
11    }
12    return 0;
13 }
```

A. 差不多是樹堆

Problem ID: treap

先讓我們回到中古時期，介紹一個小 Y 跟小 P 遇到的問題。

在當時，小 Y 跟小 P 都很喜歡蒐集各式各樣的樹，像是重心樹、後綴樹、大榕樹、線段樹、卡特蘭樹、紅黑樹、斯坦納樹等等。樹可以類比成一個 X 個點的圖透過 $X - 1$ 條無向邊連接，並且在這張圖上找不到任何一個環。然後小 Y 跟小 P 把這些收集到的大樹整理好，擺在庭院裡面種成一堆一堆的庭院佈置，並把這個庭院取名為樹堆 (Treap)，從此帶動了貴族間競相模仿的風氣，一時間家家戶戶綠意盎然。

但同時維護樹堆也帶來了很多困擾，因為有規模的設計和量產樹並不是一件簡單的事情。這一天他們得到了樹的原料，一批 N 個點以及 M 條無向帶權邊組成的連通簡單圖，點從 1 到 N 編號，邊從 1 到 M 編號。小 Y 跟小 P 想到了一個生產樹的方式：他們想要把每一張圖透過刪掉一些邊來得到樹，並使得樹上的邊權總和是所有方案中最小的，但是他們驚訝的發現，不同的圖經過這個方法產生出的樹邊權總和可能差非常多。

為了品質保證，小 Y 跟小 P 決定拿兩張圖來實驗（分別是圖 Y 和圖 P ），請你告訴小 Y 跟小 P 應該要怎麼改樹上的邊權，才可以使兩張圖經過以上的生產方式後得到的邊權總和恰好相差 K 。為了避免在改邊權的時候把整張圖弄壞，請最小化被改的邊數。

Input

測試資料第一行包含三個數字非負整數 N, M, K ，分別表示圖的點數、邊數、小 Y 和小 P 要求的邊權總和差距。

接下來 M 行描述圖 Y 的每一條邊，第 i 行有三個整數 a_i, b_i, c_i ，代表第 i 邊連結編號為 a_i 和編號 b_i 的點，並且權重為 c_i 。

接下來 M 行描述圖 P 的每一條邊，第 i 行有三個整數 d_i, e_i, f_i ，代表第 i 邊連結編號為 d_i 和編號 e_i 的點，並且權重為 f_i 。

- $1 \leq N \leq 3 \times 10^5$
- $N - 1 \leq M \leq \min(3 \times 10^5, \frac{N(N-1)}{2})$
- $0 \leq K \leq 10^9$
- $1 \leq a_i, b_i, d_i, e_i \leq N, a_i \neq b_i, d_i \neq e_i$

- 保證輸入的圖是連通簡單圖
- $|c_i|, |f_i| \leq 10^9$

Output

如果怎麼更改邊權都不能滿足小 Y 和小 P 的條件，請輸出一行 “YP can’t build a treap.” (不含引號)。否則請在第一行輸出最小需要更改的邊數 L ，接著輸出 L 行，依序代表要更動的邊。如果要把圖 Y 的第 id 條邊的邊權改成 w ，請輸出 「Y id w 」，如果要把圖 P 的第 id 條邊的邊權改成 w ，請輸出 「P id w 」，詳細的格式可以參考 Sample Output。如果有很多種更改方法，輸出任意一組皆會通過。

Note

一張圖如果

- 沒有兩條邊，它們所連接的兩個點都相同
- 每條邊所連接的是兩個不同的頂點

則稱為簡單圖。

Sample Input 1

```
3 3 2
1 2 5
2 3 6
3 1 7
1 2 5
2 3 6
3 1 3
```

Sample Output 1

```
1
P 3 4
```

Sample Input 2

```
2 1 0
2 1 0
2 1 0
```

Sample Output 2

```
0
```

B. 反分裂二元樹法

Problem ID: antisplit

NPSC 國是個團結的國家。「轉」作為該國的古老文化之一，幾乎所有 NPSC 國的程式競賽選手，在寫到平衡二元樹 (balanced binary tree) 相關的題目時都會寫伸展樹 (splay tree)。然而，總會有一些意圖破壞國家和諧的分離主義者，他們總是寫「分裂／合併式樹堆」(merge-split treap)，意圖分裂族群。為了維護二元樹的主權與領土完整，NPSC 國通過了《反分裂二元樹法》，以追蹤並調查這些傷害民族感情的人們。

NPSC 國的執法人員總共調查了 N 個分離主義者，將他們編號為 0 到 $N - 1$ ，並且發現任兩個分離主義者之間都有直接聯繫。某一天，NPSC 國的情報單位發現了一些奇怪的現象，可以描述為很多個事件，**每個事件都是這 N 個人中的某 K 個人不再與另外 $N - K$ 個人直接聯繫**。調查人員認為這可能代表著分離主義者之間發生了內鬨，而開始分門結黨、彼此分離。這件事對國家的發展有重要影響，因此調查人員把每一個這樣的事件全部記錄在一份機密文件上。

你身為 CSPN 國の間諜，輾轉取得了這份文件。這份文件經過加密，但是你成功攔截到了解密這份文件的公式，因此你決定把它解密來研究，以備不時之需。

具體來說，要解密這份文件需要一個解密函數，定義為 $F(x) = (x \text{ xor } A) \bmod N$ ，其中 A 的初始值是給定的，但會隨著解密的過程改變。($x \text{ xor } A$ 代表 x 和 A 的「按位異或」(exclusive or)，即 C/C++ 中的 x^A ； $p \bmod N$ 代表 p 除以 N 的餘數，即 C/C++ 中的 $p\%N$ 。)

這份文件解密前可以分為 M 行，每行有許多個介於 0 到 $2^{32} - 1$ 之間的非負整數，且都是下列兩種格式之一：

1. $1 \ Q \ a_1 \ a_2 \ a_3 \ \dots \ a_Q$: 代表編號為 $F(a_1), F(a_2), F(a_3), \dots, F(a_Q)$ 的人們不再與其他 $N - Q$ 個人直接聯繫。若這群人的集合為 T ，而所有人的集合為 S ，在 T 中的任兩人的聯繫狀態並不會改變，在 $S \setminus T$ 之間的任兩人亦同。一個人有可能在這個清單中出現兩次以上，但是與只出現一次是同樣的意思。保證 $1 \leq Q \leq N$ 。
2. $2 \ a \ b$: 根據編號 $F(a), F(b)$ 的人當前是否有直接聯繫改變函數 A 的值：若這兩人當前有直接聯繫，則將 A 改變為 $(22695477A + 1) \bmod 2^{32}$ ；否則，將 A 改變為 $(69069A + 1) \bmod 2^{32}$ 。假如 $F(a) = F(b)$ ，則視為兩人有直接聯繫。

這份文件是按照事件的發生順序排序，而一開始任兩個人都有直接聯繫。解密也必須按照文件的順序，也就是說遇到第二種格式時只依照所有該行之前的事件作判斷。

然而，因為這份文件實在是過於龐大，所以你決定寫一個程式來幫你解密這份文件。又因為把整份解密文件全部存下來比較費時，所以你決定先存下解密結束時的 A 值就好了。

Input

輸入的第一行有三個非負整數 N, M, A ，分別代表總人數、機密文件的行數與解密時初始的 A 值。接下來有 M 行，代表要解密的文件。詳細的格式請見題目敘述。

- $2 \leq N \leq 3 \times 10^6$
- A 與所有文件中的數字均是介於 0 和 $2^{32} - 1$ 之間的整數
- 整份要被解密的文件最多只會有 5×10^6 個數字。

Output

請輸出一行包含一個非負整數，代表整份文件解密結束之後的 A 值。

Sample Input 1

```
3 2 1
1 1 7
2 0 3
```

Sample Output 1

```
22695478
```

Sample Input 2

```
3 2 1
1 1 7
2 0 2
```

Sample Output 2

```
69070
```

C. 領地征戰

Problem ID: territorybattle

你知道「領地征戰」嗎!? 這可是當前最紅的手機遊戲呢! 並且需要兩名玩家才能進行。

在這款遊戲的世界背景中, 存在著兩大勢力—「光明會」以及「阿克西斯教」,

而兩位玩家會分別代表「光明會」的領導者以及「阿克西斯教」的領導者。

光明會(拉丁語: Illuminati), 又譯為光照派, 是 1776 年 5 月 1 日啟蒙運動時成立於巴伐利亞的一個秘密組織。

他們經常被指控合謀控制世界事務, 透過策劃事件(如法國大革命、滑鐵盧戰役和美國總統甘迺迪遇刺案),

並安插政府和企業中的代理人, 以獲得政治權力和影響力, 最終建立一個「新世界秩序」。

阿克西斯教則是一個信徒為了智障女神—阿克婭所成立宗教,

相信看過該宗教的教義之後, 即可讓人領悟阿克西斯教與其他宗教的不同之處, 以下為教義:

阿克西斯教徒只要努力就辦的到

你們都是能幹的人

就算失敗了也不是你的錯

不能成功的話就都是世界的錯

從不開心的事中逃避即可

逃避並不是失敗

有句話叫作「逃避就是勝利」

猶豫過久而得出的答案

無論如何選擇都會後悔

(既然如此) 反正都要後悔 (的話)

就選擇當下的快樂吧

別擔憂晚年會如何

未來的你是否笑著連神明都不知道

那就樂在當下吧！

好了，進入正題吧！

在這個世界中有 N 個國家 (以 1 到 N 編號) 和一些「連結著國家與國家的雙向道路」，每條道路都會有一個分數值。

一開始每個國家均不屬於兩大勢力中的任何一方，接著光明會和阿克西斯教會輪流選一個目前還不屬於兩大勢力的國家，

並對該國家進行名為教化的儀式，儀式之後這個被選的國家就會屬於他們的勢力了！

當所有國家都屬於兩大勢力的其中一方，便是要開始統計分數的時候了！對於兩方勢力分數的計算方式如下：

- 光明會分數：所有滿足「連結著兩光明會勢力國家」的道路之分數總合。
- 阿克西斯教分數：所有滿足「連結著兩阿克西斯教勢力國家」的道路之分數總合。

對於光明會的領導者來說，他會希望最終「光明會分數 - 阿克西斯教分數」最大，

而對於阿克西斯教的領導者，他會希望最終「阿克西斯教分數 - 光明會分數」最大。

眾所皆知，羅馬不是一天造成的，因為道路是一條一條蓋出來的嘛！

在遊戲剛開始時，世界中沒有任何的道路，之後會依序蓋 M 條道路，

對於每次蓋完道路後，你能告訴兩位領導人，當兩大勢力都採取最優的策略下，「光明會分數 - 阿克西斯教分數」的數值會是多少嗎？

Input

輸入的第一行有兩個正整數 N, M ，分別代表遊戲的世界中共有 N 個國家，並且之後會依序蓋出 M 條道路。接下來有 M 行，第 i 行有三個整數 a, b, c ，代表蓋了一條編號為 i 的道路，連結著編號 a 的和編號 b 的國家，並且該道路的分數值為 c 。

- $1 \leq N, M \leq 10^5$
- $1 \leq a, b \leq N$
- $-10^9 \leq c \leq 10^9$

Output

輸出共 M 行，第 i 行輸出當蓋完前 i 條路後，兩大勢力都採取最優的策略下的「光明會分數 - 阿克西斯教分數」。

Sample Input 1	Sample Output 1
5 4	0
1 2 1	1
1 3 1	1
1 4 1	2
1 5 1	

This page is intentionally left blank.

D. 分裂吧，樹堆

Problem ID: splitit

「分裂吧，樹堆」是一部在 NPSC 國很賣座的電影。電影大致的內容是：主角，NPSC 國的程式競賽選手們，在使用「分裂／合併式樹堆」(merge-split treap) 寫題目時，**寫出 bug**，**發現 bug**，**de 出 bug** 的勵志過程。

NPSC 國的程式競賽選手們使用「分裂／合併式樹堆」寫题目的狂熱程度已經無法使用文字來形容，例如最簡單的 RMQ 問題 (Range Minimum/Maximum Query，區間最大/最小值問題)，他們會直接熟練的把樹堆模板打上去，把要維護的數值好好維護一下後，就上傳了。

有一天，主角們在著名的 NOJ (NPSC Online Judge) 上面看到一道難題，看完題目之後，他們一如往常，直接把樹堆模板打上去。在對自己的程式碼自信滿滿，不測試範例測試資料的情況下，直接上傳程式碼。

結果得到 No - Wrong Answer 。

你，對於看到主角們使用樹堆上傳得到 Wrong Answer 的結果並不感到太意外，但是，你還是對主角們正在寫的難題感到興趣，題目如下：

現在 NOJ 的創辦者小 T 有 N 個樹堆，樹堆以 1 到 N 編號，第 i 個樹堆的大小為 s_i ，第 i 個樹堆裡面的數字全部都是 i 。

小 T 擁有的操作如下：

- 分裂 (split)：小 T 可以把第 i 個樹堆分裂成兩個大小分別為 a, b 的樹堆 ($1 \leq a, b \leq s_i, a + b = s_i$)，分裂第 i 個樹堆的代價是 p_i ，分裂出來的樹堆中，裡面的數字都還是 i 。這個操作對於第 i 個樹堆只能使用一次。
- 合併 (merge)：小 T 可以把兩個大小為 c 和 d 的樹堆 X, Y 合併成一個大小為 $c + d$ 的樹堆，合併之後的樹堆裡面就會有樹堆 X 裡面的數字，以及樹堆 Y 裡面的數字，合併的代價是 0。

現在，小 T 想要藉由上述的兩個操作，產生出一個大小恰好為 K 的樹堆，並且這個樹堆的數字至多出現兩種不同的數字。如果小 T 需要使用操作的話，**分裂操作一定要在合併操作之前使用完畢**。當然，如果有很多種方法可以達成的話，小 T 會希望花費的代價越少越好。

主角們對於上面的樹堆問題感到頭痛。而你認為你一定可以解出這道問題的。於是，勇敢的你，開始寫了這題，也即將掉入**寫出 bug**，**發現 bug**，**de 出 bug** 的有限輪迴之中。

Input

輸入的第一行包含兩個正整數 N, K ，代表電影的題目中，樹堆的數量，以及小 T 希望產生出來的樹堆大小。

接下來的一行，有 N 個整數，第 i 個整數為 s_i ，代表第 i 個樹堆的大小。

接下來的一行，有 N 個整數，第 i 個整數為 p_i ，代表分裂第 i 個樹堆所需要的代價。

- $1 \leq N \leq 5000$
- $1 \leq K \leq 2 \times 10^9$
- $1 \leq s_i \leq 2 \times 10^9$
- $1 \leq p_i \leq 2 \times 10^9$

Output

如果無法達成電影的題目中小 T 的心願（產生出一個大小恰好為 K 的樹堆），請輸出 -1 ，否則請輸出一個整數，代表產生出大小恰好為 K 的樹堆的最小代價。

Sample Input 1	Sample Output 1
3 7 1 2 3 4 5 6	-1

Sample Input 2	Sample Output 2
3 7 4 5 6 1 2 3	1

E. 三角形

Problem ID: triangle

在一個二維平面上，有 N 個點，保證這些點當中任取三個點皆不共線。從這 N 個點中任意選取三個相異的點，必定可以形成一個三角形。

請問：從這 N 個點中均勻隨機的選取三個相異點，所形成三角形的周長期望值是多少？

以 Sample Input 1 為例，從四個點 $(0, 0)$ 、 $(1, 0)$ 、 $(1, 1)$ 、 $(0, 2)$ 中，取三個相異點的方法總共有 4 種：

- 選取 $(0, 0)$ 、 $(1, 0)$ 、 $(1, 1)$ ：得到周長約為 3.41421356 的三角形
- 選取 $(0, 0)$ 、 $(1, 0)$ 、 $(0, 2)$ ：得到周長約為 5.23606798 的三角形
- 選取 $(0, 0)$ 、 $(1, 1)$ 、 $(0, 2)$ ：得到周長約為 4.82842712 的三角形
- 選取 $(1, 0)$ 、 $(1, 1)$ 、 $(0, 2)$ ：得到周長約為 4.65028154 的三角形

將這 4 種方法得到的三角形周長取平均後（相加再除以 4），得到答案約為 4.53224755。

Input

測試資料第一行包含一個正整數 N ，表示平面上有 N 個點。接下來 N 行，每行包含兩個正整數 X_i, Y_i ，表示第一個點的座標為 (X_i, Y_i) 。

- $3 \leq N \leq 2000$
- $-10^9 \leq X_i, Y_i \leq 10^9$
- 保證 N 個點中，任三點不共線

Output

輸出一行，包含一個浮點數，即為形成三角形的周長期望值。

如果你的答案的絕對或相對誤差不超過 10^{-6} 都會被當作正確。

Sample Input 1	Sample Output 1
4 0 0 1 0 1 1 0 2	4.5322475511

F. 十三億人都驚呆了！

Problem ID: stunned

NPSC 國是一個龐大的國家，據傳人口有十三多億人。這個國度有一個非常特別的文化：每個人都喜歡打程式競賽。因此，NPSC 國政府為了讓十三億人都驚呆，決定舉辦一場超級大規模的程式競賽，讓國內每個人都可以參加！

最後，總共有 N 個人參加了這場比賽。由於 NPSC 國政府和其請來的主辦單位非常的強大，這場比賽在如此多人參賽的情況下仍然進行地非常順利。題目一共有 M 題，採用的是 ACM 計分規則。此規則中，每題只有答對與沒有答對之分，且答對題數愈多的人，排名就愈前面（為了方便，我們假設每個人的名次皆相異，也就是沒有並列的狀況）。

比賽結束仍然處於封榜狀態時，因為參賽人數過多，顯示整個計分榜需要不少時間，主辦單位決定先公布成績分布，也就是公布對於所有的 $0 \leq i \leq M$ ，恰答對 i 題的共有幾人。

當然，在比賽的評測系統當中，計分榜其實是已經計算完畢的。然而由於評測系統的限制，主辦單位每次只能查詢某一個名次的人答對了幾題。主辦單位想要用最快的速度公布成績分布，因此希望能用很少次的查詢就算出結果。

現在請你寫一個程式和評測系統互動，並使用不超過指定次數的查詢計算出成績分布。

互動說明

首先，請由標準輸入 (standard in) 輸入兩個以空白隔開的整數 N, M ，分別代表這場比賽的參加總人數與總題數。

當你想要查詢評測系統的資料時，請輸出一行包含一個整數 x ，這個整數必須介於 1 到 N 之間，代表你想知道第 i 名的人總共答對了幾題。當你輸出完這個數字之後，記得要清空 (flush) 標準輸出 (standard out)。

當評測系統收到這個查詢之後，會把一個整數回覆到你的標準輸入 (standard in)，這個整數即是第 x 名答對的總題數。

當你算出成績分布之後，請先輸出一行包含一個“0”（不含引號），接著輸出 $M + 1$ 個整數 a_0, a_1, \dots, a_M （每個整數一行），其中 a_i 代表恰答對 i 題的人數。輸出完這些整數後，你的程式必須立刻結束 (exit)。

- $1 \leq N \leq 1\,390\,080\,000$
- $1 \leq M \leq 3\,000$
- 你的程式最多只能對評測系統做出 $\min\{61\,095, N, M \times \lceil \log_2(2N) \rceil\}$ 次查詢。如果你的程式查詢的次數超過這個數值，你的程式將會被強制結束，且被視為輸出了錯誤的答案 (Wrong Answer)。(61 095 是 NPSC 國的幸福數字。)

下列兩段範例程式分別以 C 語言、C++ 語言撰寫。這個範例程式讀入 N, M ，查詢最後一名的人答對了幾題，並回報每個人答對的題數都與最後一名相同。

```
1 #include <stdio.h>
2 int main() {
3     int N, M, i, ac_num;
4     scanf("%d%d", &N, &M);
5     printf("%d\n", N);
6     fflush(stdout);
7     scanf("%d", &ac_num);
8     puts("0");
9     for (i = 0; i <= M; i++) {
10        printf("%d\n", i == ac_num ? N : 0);
11    }
12 }
```

```
1 #include <iostream>
2 int main() {
3     int N, M, ac_num;
4     std::cin >> N >> M;
5     std::cout << N << '\n' << std::flush;
6     std::cin >> ac_num;
7     std::cout << "0\n";
8     for (int i = 0; i <= M; i++) {
9         std::cout << (i == ac_num ? N : 0) << '\n';
10    }
11 }
```

G. 爆裂遊戲

Problem ID: explosion

有一天，阿克婭和惠惠一起走在路上聊天。

「惠惠阿，妳最近是不是很久沒用爆裂魔法啦～」

「並沒有喔，我每天還是有練習喔，而且經由長期練習，我現在一天要施放 10^5 次爆裂魔法都沒有問題喔～」

「真的喔，那，讓我見識一下吧！帶妳去一個好地方。」

阿克婭帶著惠惠到了一座巨大森林面前。這座森林可以視為一個 $[-10^{18}, 10^{18}] \times [-10^{18}, 10^{18}]$ 的二維平面，每一個格子點（座標 (x, y) 中， x, y 都是整數的點）都種了一棵樹。這座森林的樹有一個特殊功能：存水。每棵樹一開始的存水量都是 0 單位，每棵樹都沒有存水量的上限，也就是說，每棵樹是可以存下無限多單位的水。

惠惠看到前面的森林是施放爆裂魔法的好地方，於是就開始準備念咒語：「比黑色更黑，比黑暗更暗的漆黑，在此寄託吾真紅的金光吧，覺醒之時的到來，荒謬教會的墮落章理，化作無形的扭曲而顯現吧，Ex-plo...」

「等一下啦，惠惠，直接讓妳釋放爆裂魔法太無聊了，為了讓妳有挑戰一點，我準備了一些東西～」

原來，身為水神的阿克婭，準備了很多的水，來挑戰惠惠的爆裂魔法。已知一單位的水需要一單位的爆裂魔法，才能讓水完全消失。

「接下來有一些指令，我們一同來完成。」阿克婭說道，「指令有倒水跟爆裂兩種。倒水指令是：我會在 (x_1, y_1) 到 (x_2, y_2) 這個線段上（包含兩個端點）所有的格子點上面的樹倒下 w 單位的水，線段不是平行於 x 軸，就是平行於 y 軸。爆裂指令是要請妳告訴我，如果妳在 (x, y) 這個格子點上，使用爆裂魔法的話，會需要使用多少單位，才能讓位於 (x, y) 的那棵樹的水完全消失。」

「不能使用爆裂魔法喔，好難過喔.....」

「如果妳成功告訴我每次爆裂指令的答案，並且都是正確的話，我就讓妳對這整座森林使用爆裂魔法。對了，當有爆裂指令出現時，妳必須要在下一個指令發生前告訴我答案喔。」

「可是我不會算數學阿.....」

你，身為惠惠的好友，決定挺身而出來幫助她。惠惠會告訴你每次阿克婭的指令，當發生爆裂指令時，請你告訴惠惠正確的答案。

Input

輸入的第一行包含一個正整數 N ，代表阿克婭的操作的數量。

接下來的 N 行，每行代表一個指令，依序是惠惠和阿克婭要完成的。指令的格式不外乎是下面兩種：

- $1\ x_1\ y_1\ x_2\ y_2\ w$ ：倒水指令，代表阿克婭會在 $(F(x_1), F(y_1))$ 到 $(F(x_2), F(y_2))$ 這個線段上所有的格子點上面的樹倒下 w 單位的水。
- $2\ x\ y$ ：爆裂指令，代表阿克婭要求惠惠算出：如果她在 $(F(x), F(y))$ 這個格子點上，使用爆裂魔法，會需要使用多少單位，才能讓位於 $(F(x), F(y))$ 的那棵樹的水完全消失。

$F(a)$ 這個函數是阿克婭設計的，用意是避免惠惠在爆裂指令之後沒有立刻回覆答案。定義函數 $F(a) = ((a + last_ans + 2 \times 10^{18} + 1) \bmod (2 \times 10^{18} + 1)) - 10^{18}$ 。其中， $last_ans$ 是上一筆爆裂指令的答案（如果之前都還沒有發生爆裂指令，則 $last_ans = 0$ ）。

- $1 \leq N \leq 10^5$
- $x_1, y_1, x_2, y_2, w, x, y$ 全部都是整數
- $-10^{18} \leq x_1, x_2, F(x_1), F(x_2) \leq 10^{18}$
- $-10^{18} \leq y_1, y_2, F(y_1), F(y_2) \leq 10^{18}$
- $-10^{18} \leq x, y, F(x), F(y) \leq 10^{18}$
- $1 \leq w \leq 10^9$
- $F(x_1) \leq F(x_2)$
- $F(y_1) \leq F(y_2)$
- $F(x_1) = F(x_2), F(y_1) = F(y_2)$ 這兩個條件至少有一個會成立

Output

對於每次爆裂指令，請輸出一個數字：如果惠惠在 $(F(x), F(y))$ 這個格子點上，使用爆裂魔法，會需要使用多少單位，才能讓位於 $(F(x), F(y))$ 的那棵樹的水完全消失。

Note

Sample Input 2 解密過後是：

2

1 *a b c d e*

2 *f g*

其中：

$$a = -10^{18} + 1$$

$$b = -10^{18} + 3$$

$$c = -10^{18} + 1$$

$$d = -10^{18} + 4$$

$$e = 1$$

$$f = -10^{18} + 1$$

$$g = -10^{18} + 4$$

Sample Input 1

5	0
2 0 0	1
1 1 0 1 2 1	2
2 1 1	
1 0 2 2 2 2	
2 1 2	

Sample Output 1**Sample Input 2**

2	1
1 1 3 1 4 1	
2 1 4	

Sample Output 2

This page is intentionally left blank.