

2017 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 16 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	A + B Problem	1
題目 B	相交在哪裏？	2
題目 C	The Jet-Black Wing ~ 漆黑之翼	4
題目 D	魔法施放距離	1
題目 E	水晶球擺設問題	1
題目 F	玩遊戲	2
題目 G	A B Problem	2

2017 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. A + B Problem

Problem ID: aplusb

A + B Problem 往往是大家在各 Online Judge 通過的第一道題目，剛學會寫程式的繪里也不例外。然而，單純地讀入兩個數字 A, B 並輸出它們的總和 C ，對於聰明伶俐又可愛的繪里來說實在太簡單了。富有挑戰精神的她忍不住想：「假如只有 A, B, C 三個數字視為字串接起來後的結果 s ，我有辦法還原出當初的數字 A, B, C ，並滿足 $A + B = C$ 嗎？」

舉例來說，如果 $s = "123"$ ，那麼可以還原出 $1 + 2 = 3$ 。注意冗餘的前導零或空字串是不被允許的，如 $s = "1023"$ ，則不能被還原成 $1 + 02 = 3$ 。

不久後，聰明伶俐又可愛的繪里就解決了這個問題。但因為這是她自己想出來的挑戰題，沒有地方可以上傳測試；所以想要請你也寫寫看，來交叉比對驗證程式的正確性。

Input

測試資料為一行包含一個字串 s 。

- $1 \leq |s| \leq 30$
- s 由數字 0123456789 組成

Output

第一行請輸出一個整數 k ，代表 s 有多少種可能的還原方法。接下來 k 行，每行請以 $A + B = C$ 的格式輸出一種還原方法。還原方法請依照 A 由小到大輸出，若 A 相同請依照 B 由小到大輸出，若 A, B 均相同請依照 C 由小到大輸出。

注意 s 可能很不幸地無法被還原，也就是可能有 $k = 0$ 的情況。

Sample Input 1	Sample Output 1
123	1 $1 + 2 = 3$
Sample Input 2	Sample Output 2
101	1 $1 + 0 = 1$
Sample Input 3	Sample Output 3
1023	0
Sample Input 4	Sample Output 4
11111122	2 $11 + 111 = 122$ $111 + 11 = 122$

B. 相交在哪裏？

Problem ID: intersection

在一個二維平面上，有 N 條直線，你想知道這些直線彼此之間是否有相交。

要表示 N 條直線彼此之間的相交關係，並不是一件容易的事情。因此，你決定先找出 K 對相交的直線來證明這 N 條直線有一定的相交數量。

Input

測試資料第一行包含兩個正整數 N, K ，表示平面上有 N 條直線，並且你需要找出 K 對相交的直線對。測試資料接下來包含 N 行，每行包含四個整數 $X_{i,1}, Y_{i,1}, X_{i,2}, Y_{i,2}$ 表示第 i 條直線通過 $(X_{i,1}, Y_{i,1})$ 及 $(X_{i,2}, Y_{i,2})$ 這兩個點。

- $2 \leq N \leq 10^5$
- $1 \leq K \leq 10^5$
- $-10^9 \leq X_{i,1}, Y_{i,1}, X_{i,2}, Y_{i,2} \leq 10^9$
- $(X_{i,1}, Y_{i,1}) \neq (X_{i,2}, Y_{i,2})$
- 保證任兩條直線不相等
- 保證這 N 條直線中至少有 K 對相交的直線

Output

請輸出 K 行，每行包含兩個整數 a_i, b_i 表示第 a_i 條直線以及第 b_i 條直線有相交。 K 對相交的直線對可以以任意順序輸出， a_i, b_i 的順序可以以任意順序輸出。如果有多組答案，請輸出任意一組。

Sample Input 1

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

Sample Output 1

```
1 2
1 3
```

Sample Input 2	Sample Output 2
3 2 0 0 0 1 0 0 1 0 0 1 1 0	1 2 2 3

C. The Jet-Black Wing ~ 漆黑之翼

Problem ID: wing

「呃啊... 可惡!... 要發狂了嗎!」

艾迪，一個自稱為「漆黑之翼」的人，正在與名為「Dark Reunion」的邪惡組織作戰。接著他就驚醒了，原來只是一場夢阿...

「我一定要變得更強。」艾迪在心中激勵自己。

為了成為一名強悍的戰鬥者，艾迪認真的鍛鍊自己。在他的練習中，他收集了 N 顆魔法石頭，第 i 顆魔法石頭有著 A_i 單位的黑暗力量。一開始，艾迪先將所有的魔法石頭依照黑暗力量的大小由小到大排序，接着，艾迪會進行 Q 次操作，每次操作有以下三種：

- 1 X：使用 X 單位的黑暗力量於每顆魔法石頭，因此，第 i 顆魔法石頭的暗黑力量會變成 $A_i \oplus X$ 單位。
- 2 L R：詢問第 L 顆魔法石頭至第 R 顆魔法石頭黑暗力量的總和。
- 3：將魔法石頭依照黑暗力量由小到大排序好。

你能夠幫助艾迪確認他操作的過程是否正確嗎？

$x \oplus y$ 表示將 x 與 y 進行互斥或 (exclusive OR) 操作。即將兩數以二進制展開後，每位元分別做互斥或運算，當兩兩數值相同時為 0，而數值不同時為 1。舉例來說， $1 \oplus 2 = 3, 2 \oplus 2 = 0$ 。這個操作存在於所有常用的程式語言中，例如：C/C++ 與 Java 即是使用「^」。

Input

測試資料第一行有兩個數字 N, Q ，表示艾迪蒐集的魔法石頭個數與訓練的操作次數。

測試資料第二行有 N 個數字 A_1, A_2, \dots, A_N ，其中 A_i 表示第 i 顆魔法石頭的黑暗力量；保證 A 序列是一個非遞減序列。

接下來 Q 行，每行表示一個操作「1 X」、「2 L R」或「3」。

- $1 \leq N, Q \leq 10^5$

- $0 \leq A_i, X < 2^{31}$
- $1 \leq L \leq R \leq N$

Output

對於每個「2 L R」的操作，輸出一個數字於一行，表示第 L 顆魔法石頭至第 R 顆魔法石頭黑暗力量的總和。

Sample Input 1

Sample Input 1	Sample Output 1
5 6	9
0 2 3 4 5	37
2 2 4	58
1 12	8
2 2 4	
3	
2 1 5	
2 1 1	

D. 魔法施放距離

Problem ID: distance

NPSC 魔法學院在今年正式成立囉！

埃迪身為 NPSC 魔法學院的第一任校長，希望能為魔法學院奠定許多良好的基礎。其中一項便是教科書的書寫，埃迪認為魔法的施放距離是十分重要的，如果施放距離不當，會產生意料不到的危險或是使得魔法效果變弱。每個魔法都有其最適合的施放距離，其表示方式有以下四種：

- `distance <= k`
- `distance < k`
- `distance >= k`
- `distance > k`

由多個魔法疊加在一起，進而產生更加強大的效果，這種魔法被稱之為鏈鎖魔法，鏈鎖魔法的最佳施放距離會是每個單一魔法最佳施放距離的交集。

身為埃迪學生的你，埃迪給了你一個作業，給你 N 個單一魔法的最佳施放距離，請求出有多少個「正整數」距離滿足「這 N 個單一魔法疊加而成的鏈鎖魔法之最佳施放距離」。

Input

測試資料的第一行包含一個正整數 N ，代表該鏈鎖魔法由 N 個單一魔法疊加而成。接下來 N 行，每行形式為以下四種中的一種: `distance <= K_i` 、`distance < K_i` 、`distance >= K_i` 、`distance > K_i` ，表示該單一魔法所適合的施放距離。

- $1 \leq N \leq 10^5$
- $1 \leq K_i \leq 10^{18}$

Output

輸出共一行，包含一個非負整數，代表有多少個「正整數」距離滿足「這 N 個單一魔法疊加而成的鏈鎖魔法之最佳施放距離」，若答案有無限多個請輸出 "INF"（不包含引號）。

Sample Input 1	Sample Output 1
2 distance < 5 distance > 3	1
Sample Input 2	Sample Output 2
1 distance > 1	INF

E. 水晶球擺設問題

Problem ID: crystal

天龍國是個擁有數千年歷史的古城，他的周遭豎立著 N 座祭壇，被編號為 1 到 N ，其中編號 x 的祭壇擁有 x 的能量值。天龍國也擁有 N 顆水晶球，一樣被編號為 1 到 N ，其中編號 y 的水晶球擁有 y 的能量值。

這些水晶球的存在，當然就是要把它們擺放到祭壇上的！

一個水晶球擺設，就是把這 N 顆水晶球放到 N 座祭壇上，使得每個祭壇上都恰好有一顆水晶球。對於一個水晶球擺設，它的擺設序列就是一個序列 a_1, a_2, \dots, a_N ，這個序列的每一項依序代表編號 i 的祭壇上擺著編號 a_i 的水晶球。

當水晶球擺設完成後，對於每一座祭壇，如果擺在它上面的水晶球的能量值比祭壇的能量值高，那這座祭壇就會發射出一道七彩的能量光束。

現在天龍國要舉行 K 天的世紀祭典，每天晚上都要擺設水晶球，讓天空中有恰好 M 道能量光束。

根據古老文獻的記載，在這 K 天的祭典中，除了第一天以外，每一天的水晶球擺設都必須比前一天的水晶球擺設還要「強壯」。

在查閱七七四十九本古書之後，你相信這裡的「強壯」就是指這個水晶球擺設的擺設序列在字典序上比前一天水晶球擺設的擺設序列還要大。更精確的說，假設某一天的擺設序列是 a_1, a_2, \dots, a_N ，而前一天的擺設序列是 b_1, b_2, \dots, b_N ，那麼必須要存在一個 $k(1 \leq k \leq N)$ ，使得對於所有 $1 \leq i < k$ ，都有 $a_i = b_i$ ，且 $a_k > b_k$ 。

已知 N, M, K 的數值，請你找出一組擺設的方法，讓辛苦的祭師能夠完成這個祭典。

Input

測試資料只有一行，當中包含三個整數 N, M, K ，依序表示水晶球（也是祭壇）的數量、祭典中需要幾道的能量光束以及祭典要舉行幾天。

- $2 \leq N \leq 1000$
- $1 \leq M < N$
- $1 \leq K \leq N - M$

Output

請輸出 K 行，每行包含 N 個正整數。這 K 行依序為第 1 天到第 K 天的水晶球擺設的擺設序列。

如果有多組解，請輸出任意一組。

可以證明，在給定的輸入條件底下，一定有至少一組解。

Sample Input 1

6 2 3

Sample Output 1

3 2 1 6 5 4
4 2 1 3 6 5
6 1 3 5 4 2

Sample Input 2

5 4 1

Sample Output 2

2 3 4 5 1

F. 玩遊戲

Problem ID: game

殿王是個天才兒童，他在六個月大的時候就已經學會乘法跟除法、一歲時學會寫程式，而現在要講的，是殿王兩歲時在玩的一款遊戲。

這是一款類似迷宮的遊戲，遊戲中有 N 個房間，編號為 0 到 $N - 1$ 。在時間 t 時，位於房間 x 的角色可以用 C_x 單位的時間移動到房間 $D_{i,t}$ ，也就是當殿王選擇「移動」這個指令後，角色便會待在房間 $D_{i,t}$ 且時間變為 $t + C_x$ 的狀態；同時殿王也可以選擇「不移動」並等待 1 單位時間，那麼就會變成角色還是待在房間 x 而時間改為 $t + 1$ 。

遊戲開始時，角色待在房間 S 、時間為 0 ，殿王需要良好的判斷，每個時間點是否要選擇「移動」，使得角色能夠在最早的時間點到達房間 Q 。

為了幫助自己破解這套遊戲，殿王把這款遊戲的執行檔拿來分析，找到了一個重要的發現：對任何一個房間 i 以及任何時間點 t ， $D_{i,t}$ 都會是 $D_{i,0} \times A^t$ 除以 N 的餘數！

殿王現在也通靈得到遊戲設置的參數了，也就是 $N, A, C_i, D_{i,0}$ ，有了這些線索之後，殿王想要有一個程式幫助他算出角色到達房間 Q 最早的時間點是多早。

以範例第一筆為例，於時間 0 時殿王可以選擇讓角色花費 6 單位的時間從房間 3 移動到房間 4，接著於時間 6 在房間 4 選擇不移動，然後於時間 7 花費 2 單位的時間從房間 4 移動到房間 1 ($3 \times 3^7 = 6561$ ，其除以 5 的餘數是 1，所以可以移動到房間 1)，便能在時間 9 時抵達目標的房間 1；但是，殿王也可以選擇在房間 3 一直等到時間 2 時再花費 6 單位時間移動到房間 1 ($4 \times 3^2 = 36$ ，其除以 5 的餘數是 1，所以這時可以移動到房間 1)，這樣就能在時間 8 時便到達目標房間了。事實上，這就是最短的時間，因此這筆測試資料的答案是 8。

Input

測試資料的第一行，包含四個整數 N, A, S, Q ，表示房間的數量、遊戲產生 $D_{i,t}$ 的參數 A 、角色初始所在的房間編號以及要前往的房間編號。接續 N 行，其中的第 $i + 1$ 行包含兩個整數 $C_i, D_{i,0}$ （也就是說，接下來的第 1 行是 $C_0, D_{0,0}$ 、第 2 行是 $C_1, D_{1,0}$ 、...、第 N 行是 $C_{N-1}, D_{N-1,0}$ ），表示從房間 i 移動到其他房間所需的時間以及房間 i 在時間 0 時可移動至的房間。

- $1 \leq N \leq 10^5$

- $0 \leq A < 10^9$
- $0 \leq S, Q < N$
- $1 \leq C_i \leq 10^5$
- $0 \leq D_{i,0} < N$

Output

輸出一行包含一個整數，即為殿王用最佳操作方式能讓角色到達目標房間的最短時間。如果不可能讓該角色到達目標房間，請輸出 -1 。

Sample Input 1	Sample Output 1
5 3 3 1 2 1 1 1 3 3 6 4 2 3	8
Sample Input 2	Sample Output 2
3 2 1 2 10 2 1 2 1 2	1
Sample Input 3	Sample Output 3
2 1 1 0 514 0 514 1	-1

G.A | B Problem

Problem ID: aorb

首殺 (First Blood) 是指在一場比賽之中，第一個對該題目獲得 **Accepted** 的隊伍稱為拿到那一題的首殺。**全場首殺**是指在一場比賽之中，第一個獲得 **Accepted** 的隊伍稱為拿到全場首殺。搶首殺一直都是比賽的樂趣之一，尤其是上傳之後顯示 Run id: 1 (全場第一次的上傳) 時拿到的首殺更讓人興奮！

因此，在比賽一開始的時候，尋找最簡單，程式碼又短的題目是搶全場首殺的關鍵！這種題目有時候可以從題目名稱下手！

瀚瀚小時候曾經參加過一場比賽，這場比賽的題本一發下來，第一道題目名字就寫著大大的「A + B Problem」！瀚瀚便很快樂地寫下以下這段程式碼，準備拿下全場首殺：

```
1 #include <iostream>
2 int main() {
3     int a, b;
4     std::cin >> a >> b;
5     std::cout << a | b << endl;
6     return 0;
7 }
```

上傳之後，果不其然顯示 Run id: 1，瀚瀚很開心的準備慶祝拿到全場首殺的喜悅，沒想到，回傳回來的結果顯示 **No, Wrong Answer!** 原來，瀚瀚因為鍵盤敲的太快，不小心將 + 敲成 |，所以整份程式碼便出錯了！

因此，瀚瀚感到很難過，但同時他也很好奇，究竟他的程式碼有多大的機率，所輸出的答案其實跟正確答案一模一樣？但這個問題對瀚瀚來說有點困難，因此，他把問題簡化為：當給定兩個正整數 A 及 X ，究竟有多少正整數 B 滿足 $B \leq X$ ，且當輸入的數字為 A 和 B 時，瀚瀚所輸出的答案會跟正確答案一樣，亦即 $A | B = A + B$ 。

$A | B$ 表示對 A, B 進行**位元或運算** (bitwise OR)，即將兩數以二進制展開後，每位元分別做 OR 運算，也就是 C/C++ 中的 | 運算子。舉例來說， $1 | 2 = 3$ ，而 $2 | 2 = 2$ ，

Input

測試資料第一行，包含一個正整數 T ，代表瀚瀚有 T 對想知道答案的 A 和 X 。接下來 T 行，每行包含兩個正整數 A, X ，代表瀚瀚所提出問題中的 A 及 X 。

- $1 \leq T \leq 10^5$
- $1 \leq A, X \leq 10^9$

Output

請輸出 T 行，每行包含一個整數代表有多少個正整數 B 滿足所給定的條件。

Sample Input 1	Sample Output 1
2 1 2 2 2	1 1