

Russia contest 10

Created: 5/3/2025 23:10

Updated: 5/3/2025 23:12

Exported: 5/3/2025 23:11

🇻🇳 Bài A. Sản xuất robot

📁 Tên file vào: chuẩn (stdin)

📁 Tên file ra: chuẩn (stdout)

🕒 Giới hạn thời gian: 1 giây

🧠 Giới hạn bộ nhớ: 256 MB

🧠 Đề bài gốc:

Trí tuệ nhân tạo GAIA, có tên là **HEPHAESTUS**, đã được phát triển mạnh mẽ và hiện đang sản xuất các cỗ máy sử dụng tài nguyên từ hệ thống mà nó được tích hợp vào. Sau khi Beta thả HEPHAESTUS vào mạng máy tính của "Thiên Đỉnh Xa", nó đã lập kế hoạch sản xuất **n** cỗ máy, trong đó cỗ máy thứ **i** cần **a_i** đơn vị tài nguyên để chế tạo.

Để tối ưu hoá, AI tìm ra cách tiết kiệm **chính xác 1 đơn vị tài nguyên cho mỗi 100 đơn vị** được sử dụng trong **mỗi** máy. Nói cách khác, sản xuất máy có chi phí **a_i** , sẽ tiết kiệm được:

$$\left\lfloor \frac{a_i}{100} \right\rfloor$$

Đơn vị tài nguyên.

💡 **Tối ưu thêm:** nếu sản xuất **hai máy i và j thành một cặp**, thì tổng tiết kiệm là:

$$\left\lfloor \frac{a_i + a_j}{100} \right\rfloor$$

Nhiệm vụ của bạn là xác định **cách ghép các cặp máy sao cho tổng tài nguyên tiết kiệm được là lớn nhất**.

⚠ Trong các phương án có **số tài nguyên tiết kiệm lớn nhất**, hãy chọn cách ghép sao cho **ít cặp nhất có thể** (tức là **nhiều máy được sản xuất đơn lẻ nhất**).

Input:

- Dòng 1: số nguyên n — số lượng máy cần sản xuất ($1 \leq n \leq 2 \cdot 10^5$)
 - Dòng 2: dãy a_1, a_2, \dots, a_n — số tài nguyên cần để sản xuất mỗi máy ($1 \leq a_i \leq 10^9$)
-

Output:

- Dòng 1: số nguyên t — số lượng tài nguyên tối đa có thể tiết kiệm.
- Dòng 2: số nguyên k — số lượng **cặp máy** cần được ghép.
- k dòng tiếp theo: mỗi dòng chứa hai số nguyên là **chỉ số** của hai máy được ghép đôi.

👉 Nếu có nhiều cách cho cùng một t và k , in ra **bất kỳ** cách nào.

Ví dụ:

Input:

```
3
30 120 190
```

Output:

```
3
1
2 3
```

Giải thích:

- Nếu sản xuất riêng:
 - $30 \rightarrow 0$
 - $120 \rightarrow 1$
 - $190 \rightarrow 1$
 $\rightarrow \text{Tổng} = 2$
- Nếu ghép 120 và 190:
 - Ghép: $(120+190)//100 = 3$
 - Máy 1 tự sản xuất: $30 \rightarrow 0$
 $\rightarrow \text{Tổng} = 3$ (tốt hơn)

Ví dụ 2:

Input:

```
6
100 4 197 324 690 500
```

Output:

```
18
2
2 3
4 5
```

Subtasks:

Subtask	Điểm	Ràng buộc thêm	Yêu cầu phụ
1	8	$n \leq 3$	—
2	14	Tất cả các a_i bằng nhau	—
3	14	$a_i \leq 50$ với mọi i	—
4	18	$n \leq 15$	—
5	20	$n \leq 1000$	Phải qua subtask 4
6	26	Không giới hạn	Qua subtask 1-5

Bạn muốn mình viết ý tưởng hoặc lời giải chi tiết cho bài này không?

Bài B. Nhật ký nhiệm vụ

 **Tên file vào:** chuẩn (stdin)

 **Tên file ra:** chuẩn (stdout)

 **Giới hạn thời gian:** 2 giây

 **Giới hạn bộ nhớ:** 256 MB

Đề bài gốc:

Eloy đi khắp nơi để tìm bản sao lưu của GAIA. Trên hành trình đó, cô gặp rất nhiều người từ các bộ tộc khác nhau, và thường được giao **các nhiệm vụ (quest)**. Để không

quên, Eloy đã tạo một **nhật ký nhiệm vụ**, hoạt động như một **hàng đợi (queue)**: nhiệm vụ nào được thêm vào trước thì sẽ được thực hiện trước.

Mỗi nhiệm vụ có một **mức độ ưu tiên** p . Ưu tiên càng **nhỏ** thì nhiệm vụ càng **quan trọng**.

Yêu cầu xử lý hai loại truy vấn:

- $+ \ t \ p$: thêm nhiệm vụ mới có **ID** là t và **ưu tiên** p vào **cuối hàng đợi**
 - $- \ -$: thực hiện **nhiệm vụ đầu hàng đợi** (in ra ID của nhiệm vụ đó)
-

Xóa nhiệm vụ không quan trọng:

Nếu một nhiệm vụ mới có ưu tiên p_0 mà **không quan trọng**, thì có thể bị xóa.

 Một nhiệm vụ được xem là **không quan trọng** nếu:

- Có ít nhất **2 nhiệm vụ trước nó** có **ưu tiên nhỏ hơn** nó.

 Khi thêm nhiệm vụ p_0 , nếu nó **không quan trọng**, Eloy kiểm tra:

- Tìm 2 nhiệm vụ đang có ưu tiên **nhỏ nhất** là p_{m1} và p_{m2}
- Tính giá trị:

$$(p_0 - p_{m1}) \cdot (p_0 - p_{m2})$$

- Nếu kết quả **lớn hơn X**, thì **xóa toàn bộ các nhiệm vụ không quan trọng**, bao gồm cả p_0 .

 Thứ tự các nhiệm vụ còn lại không thay đổi.

Input:

- Dòng 1: hai số nguyên n, X — số lượng truy vấn và hằng số kiểm tra xóa ($1 \leq n \leq 2 \cdot 10^6; 1 \leq X \leq 10^9$)
- Sau đó là n dòng:
 - $+ t p$: thêm nhiệm vụ mới có ID t , ưu tiên p ($1 \leq t, p \leq 10^9$)
 - $-$: thực hiện nhiệm vụ đầu tiên trong hàng đợi (luôn đảm bảo hàng đợi không rỗng)

Output:

- Với mỗi truy vấn $-$, in ra ID của nhiệm vụ được thực hiện (một dòng mỗi nhiệm vụ)

Ví dụ:

Input:

```
4 5
+ 1 3
+ 2 1
+ 3 5
-
```

Output:

```
1
```

Input:

```
8 5
+ 1 3
+ 2 4
+ 3 7
+ 4 3
+ 5 3
-
-
-
```

Output:

```
1
2
4
```

Subtasks:

Subtask	Điểm	Ràng buộc thêm	Yêu cầu phụ
1	7	$n \leq 8$; có tối đa 4 truy vấn '+'	—
2	15	$p_j \leq p_k$ với mọi $j < k$	—
3	15	$n \leq 100$	Subtask 1
4	13	$p_j \leq 3$ với mọi j	—
5	20	$n \leq 2000$	Subtask 3
6	20	$n \leq 10^5$	Subtask 2, 5

Subtask	Điểm	Ràng buộc thêm	Yêu cầu phụ
7	10	Không giới hạn	Subtask 1-6

Bạn có muốn mình viết thuật toán hay cài đặt mẫu cho bài này không?

🇻🇳 Bài C. Cài đặt các mô-đun GAIA

📁 **Tên file vào:** chuẩn (stdin)

📁 **Tên file ra:** chuẩn (stdout)

🕒 **Giới hạn thời gian:** 2.5 giây

🧠 **Giới hạn bộ nhớ:** 256 MB

🧠 Đề bài gốc:

Hệ thống GAIA có **n mô-đun**, mỗi mô-đun ban đầu kết nối với **slot có cùng số thứ tự**, tức là mô-đun i nối với slot i .

Có 2 thao tác **hoán vị**: p và q , mỗi thao tác có độ dài n .

- Khi thực hiện **hoán vị p** , mô-đun đang ở **slot p_i** sẽ được **chuyển đến slot i**
- Tương tự, khi thực hiện **hoán vị q** , mô-đun ở **slot q_i** sẽ đến slot i

🎯 Mục tiêu:

Chọn **cách gán mô-đun cho từng slot** bằng **hoặc là p , hoặc là $q \circ p$** (tức là thực hiện p rồi đến q), sao cho:

- Mỗi slot được gán đúng 1 mô-đun, và mỗi mô-đun chỉ gán cho 1 slot
 - Thỏa **tất cả m ràng buộc**, mỗi ràng buộc yêu cầu: *hai mô-đun a_i , b_i không được ở hai slot kế nhau*
-

Input:

- Dòng 1: hai số nguyên n, m — số mô-đun và số ràng buộc ($1 \leq n \leq 2 \cdot 10^5, 0 \leq m \leq 2 \cdot 10^5$)
 - Dòng 2: hoán vị p (gồm n số từ 1 đến n , không trùng)
 - Dòng 3: hoán vị q (tương tự)
 - Tiếp theo m dòng: mỗi dòng gồm hai số a_i, b_i — hai mô-đun không được ở **2 slot kế nhau**
-

Output:

- Nếu **không thể** chọn cách gán hợp lệ \rightarrow in -1
- Ngược lại, in ra n số nguyên:
 - Với mỗi slot i : in 1 nếu chọn gán theo p , hoặc 2 nếu chọn gán theo $q \circ p$

 Có nhiều cách thỏa mãn, in **bất kỳ**.

Ví dụ:

Input:

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

Output:

```
1 2 2
```

👉 Giải thích:

- Slot 1 chọn $p \rightarrow p_1 = 3 \rightarrow$ mô-đun 3
- Slot 2 chọn $q \circ p \rightarrow p_2 = 1, q_1 = 2 \rightarrow$ mô-đun 2
- Slot 3 chọn $q \circ p \rightarrow p_3 = 2, q_2 = 1 \rightarrow$ mô-đun 1
 \rightarrow Slot 1: mod 3, Slot 2: mod 2, Slot 3: mod 1 \rightarrow thỏa không có mod 1 và mod 3 cạnh nhau

Input:

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

Output:

```
-1
```

Subtasks:

Subtask	Điểm	Giới hạn thêm	Phụ thuộc
1	9	$n \leq 3$	—
2	14	$n \leq 18$	1
3	14	$p_i = i$ với mọi i	—
4	17	$m \leq 1$	—
5	21	$q(p(i)) = i$ với mọi i	—
6	25	Không giới hạn	1-5

Bạn có muốn mình viết lời giải hoặc ý tưởng thuật toán chi tiết cho bài này không?

Bài D. Phòng thí nghiệm dưới lòng đất

 **Tên file vào:** chuẩn (stdin)

 **Tên file ra:** chuẩn (stdout)

 **Giới hạn thời gian:** 2 giây

 **Giới hạn bộ nhớ:** 256 MB

Đề bài gốc:

Phòng thí nghiệm gồm **n phòng**, mỗi phòng nằm ở **độ sâu** $d_i > 0$ (nếu $d_i = 0$ thì nằm ở mặt đất). Trong mỗi phòng có a_i **đơn vị băng** (ban đầu là đông lạnh). Từ mỗi phòng i , có một **ống dẫn nước duy nhất** đi **xuống sâu hơn** đến phòng b_i , với $d_{b_i} > d_i$.

Duy nhất 1 phòng có $b_i = 0$, tức là nước từ đó sẽ chảy ra **hồ ngầm vô hạn**.

💧 Cách nước tan chảy và di chuyển:

- **Tại thời điểm 0 phút, băng ở các phòng nằm trên mặt đất ($d_i = 0$) tan ngay lập tức** và nước bắt đầu **di chuyển xuống**.
- Mỗi mét của ống có thể truyền **1 đơn vị nước mỗi phút**.
- Nếu nước từ phòng i đi tới b_i mất $(d_{b_i} - d_i)$ phút, thì đến thời điểm đó, **toàn bộ băng ở phòng b_i tan ra ngay lập tức** và nước bắt đầu chảy từ đó.
- Một phòng có thể nhận nước từ **nhiều ống trên đồ xuống**.

💥 Khi nước đến phòng nào đó, **băng tan ngay** và nước chảy tiếp luôn, **không có độ trễ**.

🎯 Nhiệm vụ:

Cho **m truy vấn**, mỗi truy vấn là:

- Phòng r_i , mức nước x_i .

📌 **Hỏi: phòng r_i chứa ít nhất x_i đơn vị nước trong bao nhiêu phút?**

Chỉ tính **thời gian nước ở dạng lỏng**, không tính lúc còn là băng.

📋 Input:

- Dòng 1: số nguyên n , m — số phòng và số truy vấn ($1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5$)
- Dòng n tiếp theo: mỗi dòng gồm 3 số nguyên b_i , d_i , a_i —

- b_i : nơi nước từ phòng i sẽ chảy tới (0 nếu chảy ra hồ ngầm)
- d_i : độ sâu của phòng
- a_i : số đơn vị băng
- m dòng cuối: mỗi dòng là một truy vấn gồm r_i , x_i



Output:

- Với mỗi truy vấn, in ra **số phút** mà nước trong phòng $r_i \geq x_i$
- Nếu **chưa từng đạt được**, in ra -1 .



Ví dụ:

Input:

```
3 6
3 0 2
3 0 3
0 2 1
1 2
2 4
2 1
3 1
3 2
3 3
```

Output:

```
0
0
2
```



5
3
1

Subtasks (Phân chia điểm):

Subtask	Điểm	Ràng buộc thêm	Phụ thuộc
1	16	$n, m \leq 100; d_i, a_i \leq 100$	—
2	14	Tất cả các b_i khác nhau (mỗi phòng chỉ nhận từ một phòng phía trên)	—
3	16	Mỗi phòng nhận nước từ 2 phòng , và tất cả các ống đều dài như nhau	—
4	12	$n, m \leq 100$	—
5	18	$n, m \leq 2000$	Subtask 4
6	24	Không giới hạn	1-5

Bạn có muốn mình tóm tắt chiến lược xử lý bài này hoặc viết lời giải không?