

Russia contest 7

Created: 4/29/2025 18:06

Updated: 4/29/2025 18:10

Exported: 4/29/2025 18:10

Bài A. Kế hoạch phòng thủ

Bản dịch tiếng Việt

Người Na'vi thuộc tộc Omaticaya phải bảo vệ ngôi nhà của họ – Cây Nhà – trước **n cuộc tấn công** từ "người trời", trước khi một **cuộc tổng tấn công vào Eywa** diễn ra.

Tshahìk của bộ tộc cho biết, sức mạnh của **cuộc tấn công thứ i** là a_i . Sau lần trước, các chiến binh đã luyện tập rất nhiều và có thể **chống lại các cuộc tấn công có sức mạnh không quá m** . Tuy nhiên, **nếu cuộc tấn công mạnh hơn m** , bộ tộc sẽ **chịu tổn thất**.

Jake không muốn mất một ai, nên khi có các cuộc tấn công quá mạnh, cả bộ tộc sẽ **lánh nạn trên đảo của tộc Metkayina**. Tuy nhiên, điều kiện là:

- Tộc Metkayina **chỉ cho trú ẩn đúng x cuộc tấn công liên tiếp, không ít hơn, không nhiều hơn**.
- Ngoài ra, **tộc Omaticaya phải trở về nhà ngay sau đợt tấn công thứ $n - x$** (tức là **không được đi lánh sau mốc đó**, nếu không sẽ không kịp về bảo vệ Eywa).

Jake cần tìm ra **kế hoạch tối ưu** để không có thương vong. Hãy giúp anh xác định:

- Có thể lập kế hoạch không?
 - Nếu có, **tối thiểu bao nhiêu lần** cần lánh nạn?
 - Và **đó là những thời điểm nào?**
-

Input

- Dòng 1: ba số nguyên n, x, m — số cuộc tấn công, số lần tấn công liên tiếp có thể lánh nạn, và ngưỡng sức mạnh tối đa có thể chống lại
 $1 \leq x \leq n \leq 5 \cdot 10^5, \quad 0 \leq m \leq 10^9$
- Dòng 2: n số nguyên a_1, a_2, \dots, a_n — sức mạnh của từng cuộc tấn công
 $0 \leq a_i \leq 10^9$

Output

- Nếu **không thể tránh thương vong**, in ra `-1`.
- Ngược lại, in số nguyên k — **số lần tối thiểu cần lánh nạn**, và dòng tiếp theo in k số nguyên — **chỉ số các cuộc tấn công mà trước đó cần rời đi để trú ẩn**.

Ví dụ

Input

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

Output

```
-1
```

Input

```
3 2 2
2 5 6
```

Output

1
2

Chấm điểm theo subtask

Subtask	Điểm	Giới hạn thêm	Phụ thuộc
1	18	$n \leq 10, x \leq 10$	Không có
2	13	$x = 1$	Subtask 1
3	19	$x \leq 2$	Subtask 2
4	21	$n \leq 1000$	Subtask 1
5	29	Không thêm ràng buộc	Subtask 1-4

Tóm tắt

- Có n đợt tấn công với sức mạnh a_1, \dots, a_n
- Bộ tộc chỉ chống được tấn công có $a_i \leq m$
- Nếu không chống nổi \rightarrow phải **tránh đi** đúng **x đợt liên tiếp**
- Không được tránh sau mốc đợt tấn công thứ $n - x + 1$
- Hỏi: Có thể bảo toàn lực lượng không? Nếu có, cần tránh bao nhiêu lần, tại thời điểm nào?

Bạn muốn tôi viết code giải bài này không?

Bài B. Payakan gặp nguy hiểm

Bản dịch tiếng Việt

Đại tá Quaritch cùng thuộc hạ đang truy đuổi **Payakan - một con Tulkun** - trên một con tàu khai thác. Gần đó có một căn cứ của người Na'vi, và Payakan gần như đã thoát được. Tuy nhiên, để đến được căn cứ, **Payakan phải vượt qua một khu vực rào chắn san hô**.

Khu vực này là một **lưới chữ nhật kích thước $n \times m$** , gồm các ô có thể là:

- `.` (nước, có thể bơi qua)
- `#` (san hô, không thể bơi qua)

Payakan có:

- **chiều dài k** và **chiều rộng 1** (nghĩa là nó là một đoạn thẳng dài k ô)
- Ban đầu có thể chọn bơi **nằm ngang** từ (1, 1) đến (1, k) (đầu ở (1, k)) hoặc **nằm dọc** từ (1, 1) đến (k , 1) (đầu ở (k , 1))

Nó có thể:

- Di chuyển tới/lùi theo **hướng hiện tại**
- **Xoay 90 độ** quanh đầu nó — ví dụ, từ nằm ngang chuyển thành thẳng đứng:
 - Nếu đầu ở (i , j) \rightarrow sau xoay: thân là k ô thẳng dọc hoặc ngang bắt đầu từ (i , j)
- Mỗi bước di chuyển hoặc xoay tốn **1 đơn vị thời gian**

Mục tiêu là:

- Đưa Payakan sao cho **đầu của nó nằm ở ô (n , m)**

- In ra **số bước ít nhất** cần thiết để làm điều đó, hoặc -1 nếu không thể

Input

- Dòng đầu: 3 số nguyên n, m, k — kích thước bản đồ và chiều dài của Tulkun
 $1 \leq n, m \leq 1000, 1 \leq k \leq \min(n, m), n \cdot m \leq 10^5$
- Tiếp theo là n dòng, mỗi dòng có m ký tự \cdot hoặc $\#$

Output

- Một số nguyên duy nhất — **số bước ít nhất** để Payakan tới được ô (n, m) (với đầu ở đó), hoặc -1 nếu không thể

Ví dụ

Input

```
3 3 1
...
.#.
...
```

Output

```
5
```

Input

```
6 6 3
.....#
#.###.#
#.....#
#...###
#.#.###
.#.....
```

Output

10

Input

```
2 6 2
..#...
...#..
```

Output

-1



Giải thích

- Ở ví dụ đầu tiên, Payakan có thể bắt đầu nằm ngang ở (1,1)-(1,3), tiến tới 2 bước, xoay dọc, rồi đi thêm 2 bước. Tổng cộng là **5 bước**.



Chấm điểm theo subtask



Subtask	Điểm	Giới hạn thêm	Phụ thuộc
1	20	$n, m \leq 5$	Không có
2	15	Tối đa 1 rạn san hô trên bản đồ	Subtask 1
3	15	$n \leq 2$	Subtask 1
4	15	$k = 1$	Subtask 1
5	15	$k \leq 5$	Subtask 1, 4
6	20	Không giới hạn thêm	Subtask 1-5

Tóm tắt

- Payakan là 1 đoạn thẳng dài k , bắt đầu ở góc (1,1), hướng tùy chọn
- Mỗi bước đi/lùi hoặc xoay tốn 1 đơn vị thời gian
- Mục tiêu: Đầu Payakan đến ô (n, m)
- Tránh rạn (`#`), chỉ bơi qua nước (`.`)
- In số bước nhỏ nhất hoặc -1 nếu không thể

Bạn muốn tôi viết luôn lời giải cho bài này không?

Bài C. Liên kết với Eywa

Bản dịch tiếng Việt

Các nhà khoa học nghiên cứu Pandora đã phát hiện rằng **cả hành tinh là một cơ thể sống khổng lồ**, kết nối với ý thức tập thể có tên **Eywa**. Tất cả sinh vật sống trên

Pandora đều có thể kết nối và giao tiếp với Eywa.

Để tạo ra một thế hệ **Avatar mới**, các nhà khoa học muốn **thắt chặt sự liên kết giữa Avatar và Eywa** thông qua chỉnh sửa **mã gen**.

Có tổng cộng n Avatar (trong đó n là **số chẵn**), mỗi Avatar thứ i có một **mã gen** được tạo bằng cách **xoay vòng chuỗi** s ban đầu (dài n) sang phải i **vị trí**. Nói cách khác, mã gen của Avatar thứ i là:

$$s_i, s_{i+1}, \dots, s_n, s_1, \dots, s_{i-1}$$

Tạo mã gen mới - Rekom kết hợp:

Mã gen của **thế hệ mới** được tạo bằng cách **kết hợp mã gen của hai cá thể**, bằng cách:

- Lấy **ký tự chẵn** (vị trí 0, 2, 4,...) từ cá thể thứ i
- Lấy **ký tự lẻ** (vị trí 1, 3, 5,...) từ cá thể thứ j

Cụ thể, chuỗi kết quả của cặp (i, j) sẽ là:

$$s_i, s_{(j+1) \bmod n}, s_{(i+2) \bmod n}, s_{(j+3) \bmod n}, \dots, s_{(i+n-2) \bmod n}, s_{(j+n-1) \bmod n}$$

Các nhà khoa học muốn **tìm số lượng cặp (i, j)** sao cho kết quả recombination **không trùng với bất kỳ mã gen gốc nào** của thế hệ đầu tiên (tức là **không phải hoán vị xoay của chuỗi s**).

Input

- Dòng 1: Một số chẵn n — độ dài chuỗi và số cá thể ban đầu
 $2 \leq n \leq 10^6$
- Dòng 2: Chuỗi s gồm n ký tự thường ('a'-'z')

Output

- Một số nguyên: **số lượng cặp (i, j)** (với $0 \leq i, j < n$), sao cho recombination giữa chúng tạo ra **một chuỗi mới không xuất hiện trong các xoay vòng của s**.

Lưu ý: Nếu 2 cặp khác nhau tạo cùng 1 chuỗi hợp lệ, **vẫn tính là 2 kết quả**.

Phân chia Subtask

Subtask	Điểm	Giới hạn thêm	Phụ thuộc
1	15	$n \leq 10$	Không có
2	17	Tất cả $s_i \in \{a, b\}$, số ký tự `b` ≤ 2	Subtask 1
3	17	$n \leq 100$	Subtask 1
4	23	$n \leq 1000$	Subtask 3
5	28	Không giới hạn thêm	Subtask 1-4

Ví dụ minh họa

Ví dụ 1

Input

```
4
abcd
```

Output

12

Tất cả 16 cặp (i, j), chỉ có 4 cặp sinh ra chuỗi là xoay vòng của `abcd`, còn lại 12 là mới.

Ví dụ 2

Input

4
abab

Output

8

Ví dụ 3

Input

12
aabbccaabbcc

Output

48

 Tóm tắt lại bài toán

- Có chuỗi ban đầu s dài n
 - Tạo ra n chuỗi bằng cách **xoay vòng** s
 - Tạo mã gen mới từ 2 cá thể (i, j) bằng cách xen kẽ ký tự chẵn/lẻ
 - Cần đếm có bao nhiêu cặp (i, j) **tạo ra chuỗi mới không phải là xoay vòng của s**
-

Bạn có muốn tôi viết code giải bài này bằng C++ hoặc Python không?

Bài D. Đột kích vào tàu vận chuyển

Bản dịch tiếng Việt

Sau khi "người trời" quay lại Pandora và xây dựng căn cứ mới, **Jake Sully** thường xuyên tổ chức **các cuộc đột kích** vào **tàu vận chuyển tài nguyên và vũ khí**.

Tuy nhiên, vì vũ khí của loài người quá mạnh , mỗi cuộc đột kích cần được **chuẩn bị kỹ lưỡng**.

Mô tả đội hình đột kích

Có n chiến binh Na'vi (thuộc tộc Omaticaya) sẵn sàng tham gia, mỗi người có:

- **Sức mạnh:** a_i
- **Tốc độ bay của Ikran:** b_i

Jake muốn chọn một **đội hình gồm một hoặc nhiều người**, theo **thứ tự tấn công** sao cho:

1. Sức mạnh không giảm:

$$a_{i_1} \leq a_{i_2} \leq \dots \leq a_{i_k}$$

2. Tốc độ không giảm nhẹ và chênh lệch không quá x :

$$b_{i_1} \leq b_{i_2} \leq \dots \leq b_{i_k} \leq b_{i_1} + x$$

💡 Nghĩa là: các chiến binh càng về sau càng mạnh hơn (hoặc bằng), tốc độ phải gần nhau (tối đa lệch x) để giữ đội hình chặt chẽ trong khi bay.”

Jake còn có **quyền gọi thêm một chiến binh mới**, với **sức mạnh và tốc độ tùy chọn** trong đoạn $[1, 10^9]$.

🎯 Mục tiêu của bài toán

Bạn hãy giúp Jake:

- **Chọn thông số (a, b)** cho **người mới** sao cho:
 - Khi thêm người này vào danh sách, **đội hình có độ dài lớn nhất có thể**
- In ra:
 - **Độ dài đội hình dài nhất**
 - **Cặp (a, b)** cho người mới giúp đạt được điều đó

📥 Input

- Dòng 1: Hai số nguyên n, x — số người ban đầu và độ lệch tốc độ tối đa
 $1 \leq n \leq 2 \cdot 10^5, \quad 0 \leq x \leq 10^9$
- n dòng tiếp theo: mỗi dòng chứa 2 số nguyên a_i, b_i — sức mạnh và tốc độ



Output

- Dòng 1: Số nguyên — **độ dài đội hình lớn nhất**
- Dòng 2: Hai số nguyên — giá trị a, b của người mới nên được thêm vào



Ví dụ minh họa

Ví dụ 1

Input

```
1 5
3 3
```

Output

```
2
3 3
```



Người cũ và người mới có cùng thông số \Rightarrow thỏa mãn điều kiện

Ví dụ 2

Input

```
3 3
1 2
```





2 5
4 11

Output

4
2 8

➡ Chọn người mới có sức mạnh trung bình và tốc độ trung bình \Rightarrow kết nối được các nhóm có sẵn

Chấm điểm theo Subtask

Subtask	 Điểm	 Giới hạn thêm	 Phụ thuộc
1	12	$n \leq 10$	Không có
2	13	$x = 0$, và $a_i = b_i$	Subtask 1
3	17	$x = 0$	Subtask 2
4	17	Tất cả a_i giống nhau	Subtask 1
5	19	$n \leq 1000$	Subtask 1
6	22	Không giới hạn thêm	Subtask 1-5

Tóm tắt bài toán

- Có n người, mỗi người có (a_i, b_i)

- Tìm 1 người mới (a, b) sao cho khi thêm vào:
 - Chuỗi người có:
 - Sức mạnh tăng dần
 - Tốc độ gần nhau (lệch không quá x)
 - In ra:
 - Độ dài đội hình lớn nhất có thể
 - Cặp (a, b) để đạt được điều đó
-

Bạn muốn mình giải bài toán này bằng C++ hoặc Python không? 😊