

Russia contest 10

Created: 5/3/2025 00:01
Updated: 5/3/2025 00:03
Exported: 5/3/2025 00:02

Bài A. Trở nên mạnh mẽ hơn

Tên file đầu vào: tiêu chuẩn

Tên file đầu ra: tiêu chuẩn

Giới hạn thời gian: 1 giây

Giới hạn bộ nhớ: 256 MB

Bối cảnh:

Aiden Caldwell đang sống sót trong một thế giới đầy xác sống. Để tăng cường tạm thời khả năng của mình, anh ta sử dụng một **chất ức chế** gồm `n` thành phần.

Thiết bị kích hoạt:

Là một **ngăn xếp (stack)**, cho phép:

- Đầu tiên: thêm tùy ý số lượng thành phần vào.
- Sau đó: lấy các thành phần ra **theo thứ tự ngược lại** (LIFO - Last In First Out).

Lưu ý quan trọng:

- Sau khi đã **lấy ra bất kỳ thành phần nào**, thì **không thể thêm mới vào nữa**, chỉ được phép tiếp tục lấy ra.

Điều kiện để chất ức chế hoạt động:

1. Thành phần thứ `i` phải nằm trong thiết bị **chính xác** `a_i` giây.
2. **Giữa hai lần đưa vào** liên tiếp phải cách nhau **ít nhất 1 giây**.
3. **Giữa hai lần lấy ra** liên tiếp cũng phải cách nhau **ít nhất 1 giây**.

Vấn đề đặt ra:

Một thiết bị đôi khi **không đủ**, vì các ràng buộc thời gian khiến việc thêm/lấy không thể thỏa mãn toàn bộ. Khi đó cần sử dụng **nhiều thiết bị**.

Mục tiêu:

Tìm **số lượng thiết bị ít nhất cần dùng** để có thể sử dụng toàn bộ n thành phần sao cho chất ức chế hoạt động đúng yêu cầu.

Input:

- Dòng đầu tiên: một số nguyên n — số lượng thành phần ($1 \leq n \leq 2 \cdot 10^5$).
- Dòng thứ hai: dãy a_1, a_2, \dots, a_n — thời gian mỗi thành phần cần nằm trong thiết bị ($1 \leq a_i \leq 10^9$).

Output:

- In ra một số nguyên — **số thiết bị tối thiểu cần thiết**.

Subtasks:

Phân đoạn	Điểm	Giới hạn bổ sung	Phụ thuộc	Mức chấm điểm
1	12	$n \leq 3$	-	đầy đủ
2	15	$n \leq 7$	1	đầy đủ
3	15	$a_i < a_{i+1}$	-	đầy đủ

Phân đoạn	Điểm	Giới hạn bổ sung	Phụ thuộc	Mức chấm điểm
4	18	a_i là số chẵn	-	đầy đủ
5	20	$n \leq 1000$	2	đầy đủ
6	20	Không giới hạn gì thêm	1-5	chấm theo lỗi đầu tiên

Ví dụ:

Input 1:

```
2
1 2
```

Output 1:

```
2
```

Input 2:

```
3
3 6 1
```

Output 2:

```
1
```

Input 3:

5
1 2 4 3 2

Output 3:

3

Giải thích ví dụ 3:

- Thiết bị 1: chứa thành phần số 4 và 1 (thời gian đủ và cách nhau đúng quy tắc).
- Thiết bị 2: chỉ chứa thành phần số 2.
- Thiết bị 3: chứa thành phần số 3 và 5. Thứ tự vào và ra đảm bảo đúng khoảng cách thời gian.

Bài B. Chỉ số sinh học (Biomarkers)

Tên file đầu vào: tiêu chuẩn

Tên file đầu ra: tiêu chuẩn

Giới hạn thời gian: 1 giây

Giới hạn bộ nhớ: 256 MB

Bối cảnh:


Ở thành phố **Villedor**, mỗi người dân đều có một **thiết bị sinh học (biomarker)** cho biết mức độ **nhiễm virus zombie**.

- Càng **lớn** thì người khác càng **tránh xa**, vì **nguy cơ phát bệnh zombie cao** hơn.

- Một người **sẽ bị phát bệnh** nếu **chỉ số sinh học chia hết cho 3**.

 Tuy nhiên, thiết bị của **Aiden** bị lỗi: nó có thể hiển thị **thừa chữ số**.

Ví dụ: chỉ số thực tế là ``123``, thiết bị có thể hiển thị ``1234``, ``19203``, nhưng **không** thể hiển thị ``1453`` (vì không thể tạo ``1453`` bằng cách chèn số vào ``123`` theo thứ tự giữ nguyên).

 Aiden **biết chắc chắn** rằng chỉ số thật của mình **chia hết cho 3**, và muốn tìm giá trị **lớn nhất** có thể là **chỉ số thật** (tức là một **dãy con** của ``n`` chia hết cho 3).

Input:

- Dòng đầu tiên: số nguyên k — số lượng chữ số của số đang hiển thị ($1 \leq k \leq 5 \cdot 10^5$).
- Dòng thứ hai: k chữ số (``0`` đến ``9``), không có số 0 đứng đầu.

Output:

- In ra **giá trị lớn nhất chia hết cho 3** có thể là **chỉ số thật của Aiden**, bằng cách **xoá một số chữ số nào đó** khỏi số hiển thị hiện tại.
- Nếu **không có dãy con nào chia hết cho 3**, in ra ``0``.

Subtasks:

Phân đoạn	Điểm	Giới hạn bổ sung	Phụ thuộc	Mức chấm
1	8	$k \leq 3$	-	đầy đủ
2	13	$k \leq 6$	1	đầy đủ
3	15	$k \leq 18$	2	đầy đủ
4	18	$k \leq 350$, số n không chứa số 0	3	đầy đủ
5	12	Các chữ số trong n không giảm	-	đầy đủ
6	15	n không chứa số 0	1-5	chấm lỗi đầu tiên
7	19	Không giới hạn	6	chấm lỗi đầu tiên

Ví dụ:

Input 1:

4
1234

Output 1:

234

Input 2:

5
54784

Output 2:

5784

Input 3:

2
80

Output 3:

0

Giải thích ví dụ 1:

Các dãy con có thể tạo ra:

- `3`, `12`, `24`, `123`, `234`, ...

Trong đó `234` là dãy con chia hết cho 3 lớn nhất.

Bài C. Di chuyển tự do (Свободное перемещение)

Tên file đầu vào: tiêu chuẩn

Tên file đầu ra: tiêu chuẩn

Giới hạn thời gian: 2 giây

Giới hạn bộ nhớ: 256 MB

Bối cảnh:

Aiden Caldwell là một bậc thầy parkour sống trong thành phố **Villedor**, nơi có:

- n **địa điểm** (node)
- m **đường đi hai chiều** giữa các địa điểm (edge)

🚀 Để tối ưu di chuyển, Aiden có thể:

- Đặt **thiết bị hỗ trợ** giúp di chuyển nhanh **chỉ theo một chiều** của mỗi đoạn đường (phải chọn đúng 1 chiều cho mỗi đoạn).

🔄 **Một bộ ba ($a \rightarrow b \rightarrow c$)** được gọi là **tiện lợi** nếu:

- Có thể **di chuyển nhanh** từ $a \rightarrow b$ và từ $b \rightarrow c$.

🎯 **Mục tiêu:**

Chọn hướng di chuyển nhanh cho mỗi đoạn đường sao cho **tổng số bộ ba tiện lợi là lớn nhất có thể**.

📥 **Input:**

- Dòng 1: hai số nguyên n, m ($2 \leq n \leq 3 \cdot 10^5$, $1 \leq m \leq 3 \cdot 10^5$)
- m dòng sau: mỗi dòng gồm hai số u_i, v_i — thể hiện một đoạn đường giữa hai địa điểm (không có đoạn nào trùng nhau).

📤 **Output:**

- Dòng đầu: một số nguyên — **tổng số bộ ba tiện lợi lớn nhất**.
- m dòng tiếp theo: mỗi dòng gồm hai số x_i, y_i , nghĩa là hướng di chuyển nhanh là từ x_i đến y_i .

👉 Bạn có thể in ra thứ tự các cạnh **khác với thứ tự trong input**.

👉 Nếu có nhiều cách đạt tối đa, in **bất kỳ cách nào hợp lệ**.

Subtasks:

Phân đoạn	Điểm	Giới hạn bổ sung	Phụ thuộc	Mức chấm
1	18	$m \leq 20$	-	đầy đủ
2	18	Mỗi địa điểm nối với tối đa 2 địa điểm khác	-	đầy đủ
3	20	Đồ thị đầy đủ: $m = \frac{n(n-1)}{2}$	-	đầy đủ
4	16	Đồ thị là cây (chỉ có một đường giữa mọi cặp đỉnh)	-	đầy đủ
5	28	Không giới hạn	1-4	chấm lỗi đầu tiên

Ví dụ:

Input 1:

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

Output 1:

```
1
1 2
3 1
```

Input 2:

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

Output 2:

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



Giải thích ví dụ 2:

Với cách chọn hướng như trên, các bộ ba tiện lợi gồm:

- `(1 → 2 → 3)`
- `(2 → 3 → 1)`
- `(3 → 1 → 2)`
- `(1 → 4 → 3)`

- `(4 → 3 → 1)`
 - `(3 → 1 → 4)`
- Tổng cộng: **6 bộ ba tiện lợi.**

💣 Bài D. Phá hủy cối xay gió (Подрыв ветряка)

Tên file đầu vào: tiêu chuẩn

Tên file đầu ra: tiêu chuẩn

Giới hạn thời gian: 1 giây

Giới hạn bộ nhớ: 256 MB

💣 Bối cảnh:

Aiden đứng về phe **Người sống sót** và nhận nhiệm vụ **làm suy yếu cối xay gió** của phe **Peacekeepers**.

- Cối xay gió có **độ ổn định ban đầu** là s .
- Có **n bộ phận quan trọng**. Bộ phận thứ i :
 - Chỉ có thể **vô hiệu hóa** khi độ ổn định hiện tại **không nhỏ hơn** a_i .
 - Sau khi vô hiệu hóa, độ ổn định sẽ **thay đổi thêm** b_i (có thể dương hoặc âm).

📈 Yêu cầu:

- Aiden có thể **tùy ý chọn** các bộ phận có thể vô hiệu hóa tại thời điểm hiện tại.
- Không cần phải tắt hết tất cả.
- Aiden có thể **dừng lại bất kỳ lúc nào**.
- Độ ổn định cuối cùng **có thể âm**.

🎯 Mục tiêu:

- Tìm cách vô hiệu hóa một số bộ phận (có thể là 0), theo một thứ tự, sao cho:
 - Độ ổn định cuối cùng là **nhỏ nhất có thể**.
 - In ra thứ tự tắt các bộ phận.
-



Input:

- Dòng 1: hai số nguyên n, s — số bộ phận và độ ổn định ban đầu ($1 \leq n \leq 1000$, $0 \leq s \leq 10^4$).
- n dòng sau: mỗi dòng gồm hai số a_i, b_i ($0 \leq a_i \leq 2 \cdot 10^4$, $-10^4 \leq b_i \leq 10^4$) — điều kiện truy cập và thay đổi độ ổn định khi vô hiệu hóa bộ phận thứ i .

✂ Tổng $\sum |b_i| \leq 2 \cdot 10^4$



Output:

- Dòng 1: hai số ans, k — độ ổn định cuối cùng nhỏ nhất có thể và số lượng bộ phận cần tắt.
- Dòng 2: k số — chỉ số của các bộ phận theo thứ tự thực hiện.

Nếu có nhiều cách cho kết quả tốt nhất, in ra **bất kỳ cách nào hợp lệ**.



Subtasks:



Phân đoạn	Điểm	Giới hạn bổ sung	Phụ thuộc	Mức chấm
1	8	$n \leq 8$	-	đầy đủ
2	10	$n \leq 20$, tất cả $b_i < 0$	-	đầy đủ
3	12	Chỉ có một i sao cho $b_i < 0$	-	đầy đủ
4	12	Tất cả $a_i = 0$	-	đầy đủ
5	16	Mọi $b_i < 0$ đều bằng nhau	2, 3	đầy đủ
6	20	Có tối đa 20 phần tử với $b_i < 0$	1 - 3	chấm lỗi đầu tiên
7	22	Không giới hạn gì thêm	1 - 6	chấm lỗi đầu tiên

Ví dụ:

Input 1:

```
3 10
10 -2
10 6
15 -9
```

Output 1:

```
7 2
2 3
```

Input 2:

```
5 100
180 20
100 79
```

```
179 -80
180 -90
1 1
```

Output 2:

```
90 3
5 2 4
```

Input 3:

```
3 50
50 -30
30 -40
40 -20
```

Output 3:

```
-10 2
3 2
```

Giải thích ví dụ 3:

- Bắt đầu từ 50
- Vô hiệu hóa bộ phận 3: $50 - 20 = 30$
- Vô hiệu hóa bộ phận 2: $30 - 40 = -10$
 - Không thể thực hiện thêm vì `a_i` của bộ phận 1 là 50, nhưng hiện tại đã nhỏ hơn 50.
 - Kết quả tốt nhất là -10.