

# Russia contest 3

Created: 4/25/2025 13:00

Updated: 4/25/2025 13:02

Exported: 4/25/2025 13:03



## Dịch đề bài:

### Bài A. Tam giác

**Giới hạn thời gian:** 1 giây

**Giới hạn bộ nhớ:** 256 MB

Hình học yêu thích của các con gái Gru là **tam giác**. Khi rảnh rỗi (ngoài giờ học và giúp Gru làm việc xấu), các bé thích chơi trò sau:

- Mỗi bé chọn một số nguyên dương.
- Sau đó, cả nhóm cùng kiểm tra xem có tồn tại **tam giác không suy biến** (tam giác có diện tích dương) với ba cạnh là ba số đó hay không.

Một lần, các bé tìm thấy một tập hợp gồm  **$n$  số nguyên dương**  $a$ . Các bé muốn biết có bao nhiêu số nguyên  **$x$**  khác, sao cho  **$x$  và hai số bất kỳ khác** từ tập hợp  $a$  **luôn tạo thành một tam giác không suy biến**.

Hãy giúp các bé giải bài toán này.

---

## Input:

- Dòng đầu tiên: số nguyên  $n$  ( $2 \leq n \leq 5 \cdot 10^5$ ).
- Dòng thứ hai:  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

## Output:

- Một số nguyên: số lượng giá trị  $x$  thỏa mãn.
- 

## Giải thích ví dụ:

### Ví dụ 1:

3  
3 3 5

- Các cặp chọn từ tập  $a$ : (3,3), (3,5), (3,5).
- Các giá trị  $x$  có thể chọn là: 3, 4, 5 (3 giá trị).

### Ví dụ 2:

3  
3 1 2

- Không tồn tại giá trị  $x$  nào thỏa mãn.

### Ví dụ 3:

5  
9 5 6 7 9

- Các giá trị  $x$  có thể chọn là: 3, 4, 5, 6, 7, 8 (6 giá trị).
- 

## Tóm tắt đề bài:

- Cho dãy số  $a$ .

- Tìm số lượng giá trị nguyên  **$x$**  sao cho với **mọi cặp hai phần tử bất kỳ** từ  $a$ , ba cạnh  $x$ , phần tử 1, phần tử 2 **tạo thành một tam giác không suy biến**.

## Subtasks (các phần nhỏ):

Phần	Điểm	Giới hạn	Yêu cầu phụ thuộc
0	-	Chạy ví dụ trong đề	-
1	14	$n \leq 500, a_i \leq 200$	-
2	15	$n \leq 2000, a_i \leq 500$	0, 1
3	11	$a_i \leq 4$	0, 1
4	13	Tất cả $a_i = a_j$	0, 1
5	14	$a_1 = 1$	0, 1
6	16	$a_i \leq 2 \cdot 10^5$	0, 1, 2, 3
7	17	Không giới hạn	0 – 6



## Dịch đề bài:

### Bài B. Phá khóa kết sắt

**Giới hạn thời gian:** 2 giây

**Giới hạn bộ nhớ:** 256 MB

Gru cần **phá khóa kết sắt** của Vector – nơi chứa bản thiết kế tên lửa cho nhiệm vụ **cướp Mặt Trăng**. Đây là nhiệm vụ phức tạp, không thể giao cho Minion, nên bạn phải giúp Gru giải quyết.

Trên màn hình của **khóa mã** hiện tại hiển thị một chuỗi **s** gồm các chữ cái thường trong bảng chữ cái tiếng Anh. Chuỗi **t** (cùng độ dài với s) là **từ yêu thích** của Vector, và Gru chắc chắn kết sẽ mở khi chuỗi trên màn hình biến thành **t**.

Bên cạnh màn hình có một **ô nhập số nguyên** và một **nút bấm** duy nhất. Theo nghiên cứu của tiến sĩ Nefario, bạn có thể:

1. Nhập vào một số nguyên **x** từ **1 đến n** (với  $n$  là độ dài chuỗi s).
2. Khi bấm nút, chuỗi **s** sẽ chia thành hai phần:
  - **Tiền tố**  $p$  gồm  $n - x$  ký tự đầu.
  - **Hậu tố**  $q$  gồm  $x$  ký tự cuối.
3. Sau đó, chuỗi **s** sẽ bị thay thế bởi **q\_rev + p** (tức là đảo ngược hậu tố q rồi nối với tiền tố p).

#### Ví dụ:

Với chuỗi **s = "arkshs"** và nhập **x = 2**:

- Tiền tố  $p = "arks"$ , hậu tố  $q = "hs"$ .
- Đảo ngược q:  $q_{rev} = "sh"$ .
- Kết quả:  $s = "sharks"$ .

Gru không muốn gây nghi ngờ, nên **số lần bấm nút không được vượt quá m**. Hãy tìm cách biến **s thành t** trong **không quá m lần bấm nút**.

---

#### Input:

- Dòng đầu: hai số nguyên  $n$  và  $m$  ( $1 \leq n \leq 2000; 5100 \leq m \leq 10^4$ ).
- Dòng tiếp: chuỗi **s** (gồm  $n$  chữ cái thường).

- Dòng kế: chuỗi **t** (gồm n chữ cái thường).

## Output:

- Nếu **không thể** biến s thành t sau tối đa **m lần bấm nút**, in ra **-1**.
- Ngược lại:
  - Dòng đầu in số lần bấm nút **k** ( $0 \leq k \leq m$ ).
  - Dòng sau in **k số nguyên**, mỗi số là giá trị **x** đã nhập ở mỗi lần bấm nút.

## Lưu ý:

- **Không yêu cầu tối thiểu số lần bấm nút**, chỉ cần không vượt quá **m**.

---

## Giải thích ví dụ:

### Ví dụ 1:

```
6 10000
abacbb
babcba
```

- Các bước biến đổi:
  1. abacbb → bbcaba (x = 6)
  2. bbcaba → ababbc (x = 3)
  3. ababbc → cbabab (x = 2)
  4. cbabab → babcba (x = 3)

### Ví dụ 2:

3 10000

aba

bba

- Không thể biến  $s$  thành  $t \rightarrow$  in ra **-1**.

## Tóm tắt đề bài:

- Cho hai chuỗi  $s$  và  $t$  (cùng độ dài).
- Mỗi lần, được **đảo ngược đoạn cuối của  $s$  dài  $x$  ký tự rồi đưa lên đầu**.
- Tìm cách biến  $s$  thành  $t$  sau tối đa  $m$  lần thao tác.

## Subtasks (các phần nhỏ):

Phần	Điểm	Giới hạn	Yêu cầu phụ thuộc
0	-	Chạy ví dụ trong đề	-
1	9	$n \leq 8, m = 10^4$	0
2	21	$n \leq 100, m = 10^4$	0, 1
3	24	$n \leq 1000, m = 10^4$	0 - 2
4	12	$m = 10^4$	0 - 3
5	12	$m \leq 8100$	0 - 4
6	11	$m \leq 6100$	0 - 5
7	11	Không giới hạn	0 - 6



## Dịch đề bài:

### Bài C. Bức tranh lục giác

**Giới hạn thời gian:** 3 giây

**Giới hạn bộ nhớ:** 512 MB

Kevin (một Minion) rất thích **vẽ trên lưới hình lục giác**. Một **lưới lục giác** là lưới chia mặt phẳng thành các hình lục giác đều bằng nhau.

Mỗi lục giác được xác định bởi một cặp tọa độ nguyên  $(x, y)$ , trong đó:

- Trục hoành hướng **xuống dưới**.
- Trục tung nghiêng **120 độ** so với trục hoành.

**Hai ô lục giác** được gọi là **láng giềng** nếu chúng **có chung một cạnh** (ví dụ: ô  $(1,3)$  và  $(2,2)$  là láng giềng; còn  $(1,3)$  và  $(3,2)$  thì không).

Một tập hợp các ô được gọi là **liên thông** nếu từ **bất kỳ ô nào** cũng có thể di chuyển đến **bất kỳ ô nào khác** trong tập, chỉ qua các ô láng giềng.

Tập hợp đó được gọi là **hai lần liên thông** nếu giữa **bất kỳ hai ô** đều tồn tại **ít nhất hai đường đi** (không trùng nhau về ô đi qua).

Hôm nay, **Stuart và Bob** đã tô màu một số ô (không nhất thiết liên thông).

Kevin cho rằng bức tranh này **đẹp** nếu **không có ba ô láng giềng cùng tô màu**, tức là **không có cụm tam giác**.

Kevin muốn **xóa bớt các ô** tô màu sao cho **không còn cụm tam giác nào**. Nhưng anh ấy muốn **xóa ít ô nhất có thể** để **giữ lại ý tưởng ban đầu** của Stuart và Bob.

**Nhiệm vụ của bạn:** Tìm các ô cần xóa để bức tranh **không còn cụm tam giác**.

- **Không bắt buộc phải tối ưu số ô xóa**, nhưng **xóa càng ít ô thì điểm càng cao**.



## Input:

- Dòng đầu: ba số nguyên  $n$ , **cost**, và **threshold**:
  - $n$  là số ô tô màu ( $1 \leq n \leq 10^5$ ).
  - cost** và **threshold** dùng để **chấm điểm**, bạn **không cần quan tâm**.
- Dòng tiếp theo:  $n$  cặp tọa độ  $(x_i, y_i)$ , mỗi cặp trên một dòng ( $|x_i|, |y_i| \leq 10^9$ ).
  - Không có cặp nào trùng lặp.

## Output:

- Dòng đầu: số ô bạn xóa **k**.
- Tiếp theo: **k dòng**, mỗi dòng là tọa độ ô bị xóa.

## Lưu ý:

- Nếu bạn xóa **ô không tô màu** hoặc **vẫn còn cụm tam giác** sau khi xóa → **WA** (sai).
- Cụm tam giác** là **3 ô láng giềng** tạo thành một hình tam giác.

## Giải thích ví dụ:

### Ví dụ 1:

```
3 0 0.0
1 0
0 1
1 1
```

- 3 ô tạo thành một tam giác → phải xóa ít nhất 1 ô.



- Xóa ô (1,0) là đủ.

### Ví dụ 2:

```
7 0 0.0
1 2
0 2
1 1
2 1
2 2
1 3
0 3
```

- Có cụm tam giác → chỉ cần xóa ô (1,2).

### Tóm tắt đề bài:

- Cho danh sách các ô lục giác đã tô màu trên **lưới lục giác vô hạn**.
- Tìm các ô cần **xóa ít nhất có thể** sao cho **không còn 3 ô láng giềng** nào tạo thành **tam giác**.

### Subtasks (các phần nhỏ):

Phần	Điểm	Giới hạn	Giải thích thêm
1-5	2	Không có hạn chế	-
6-10	3	$n \leq 3$	-
11-20	2	$n \leq 18$	-

Phần	Điểm	Giới hạn	Giải thích thêm
21-30	2	$n \leq 1000$ , tập hai lần liên thông	-
31-39	3	Mỗi ô có <b>không quá 3 láng giềng</b>	-
40-43	5	Không hạn chế	-

### Cách chấm điểm:

- Nếu số ô bạn xóa **nhiều hơn gấp đôi** so với lời giải ban tổ chức → **0 điểm**.
- Nếu ít hơn → tính điểm dựa trên **tỷ lệ so với lời giải ban tổ chức**.



### Dịch đề bài:

#### Bài D. Chuyến du lịch của Minion

**Giới hạn thời gian:** 1 giây

**Giới hạn bộ nhớ:** 256 MB

Các Minion đã **du lịch rất nhiều** để tìm kiếm một **thủ lĩnh mới**, trước khi gặp Gru. Có  **$n$  thành phố** được xem là **tiềm năng nhất**, với thành phố thứ  $i$  có **độ tiềm năng** là  $a_i$ .

Tuy nhiên, Minion **không chỉ quan tâm đến độ tiềm năng**, mà còn muốn có **trải nghiệm vui vẻ**. Mỗi thành phố  $i$  có thể được ghé qua theo **2 cách**:

1. **Đi qua** (không dừng lại): họ nhận được **niềm vui** là  $(a_i - c)^2$ .
2. **Ở lại 1 ngày**: họ nhận được **niềm vui** là  $(a_i - a_{\text{prev}})^2$ , trong đó  $a_{\text{prev}}$  là độ tiềm năng của **thành phố cuối cùng mà họ đã ở lại 1 ngày**.

- **Lưu ý đặc biệt:**

- Thành phố **đầu tiên** mà họ ở lại sẽ có  $a_{\text{prev}} = a_i$ , và **niềm vui nhận được bằng 0** tại thành phố đó.

Hành trình bắt buộc phải:

- Thăm tất cả các thành phố theo thứ tự từ 1 đến n.**
- Ở lại 1 ngày tại thành phố 1 và thành phố n** (không được đi qua).

### Nhiệm vụ của bạn:

Giúp Minion tính **tổng niềm vui tối đa** mà họ có thể đạt được trong hành trình.

---

### Input:

- Dòng đầu tiên: hai số nguyên  $n$  và  $c$  ( $1 \leq n \leq 10^6; 10^6 \leq c \leq 10^6$ ).
- Dòng tiếp theo:  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $10^6 \leq a_i \leq 10^6$ ) là độ tiềm năng của các thành phố.

### Output:

- Một số nguyên: tổng **niềm vui tối đa** có thể đạt được.

---

### Giải thích ví dụ:

#### Ví dụ 1:

```
6 3
5 1 6 5 0 1
```

- Bắt buộc **ở lại thành phố 1 (5)**, niềm vui = 0.

- Thành phố 2: chọn **đi qua**, niềm vui =  $(1 - 3)^2 = 4$ .
- Thành phố 3: chọn **ở lại**, niềm vui =  $(6 - 5)^2 = 1$ .
- Thành phố 4: **đi qua**, niềm vui =  $(5 - 3)^2 = 4$ .
- Thành phố 5: **đi qua**, niềm vui =  $(0 - 3)^2 = 9$ .
- Thành phố 6: **ở lại**, niềm vui =  $(1 - 6)^2 = 25$ .

Tổng niềm vui =  $0 + 4 + 1 + 4 + 9 + 25 = 43$ .

## Tóm tắt đề bài:

- **Thăm n thành phố theo thứ tự**, mỗi thành phố có thể:
  - **Đi qua** → niềm vui:  $(a_i - c)^2$ .
  - **Ở lại 1 ngày** → niềm vui:  $(a_i - a_{\text{prev}})^2$ .
- **Bắt buộc ở lại thành phố 1 và thành phố n.**
- Tìm cách chọn sao cho **tổng niềm vui tối đa**.

## Subtasks (các phần nhỏ):

Phần	Điểm	Giới hạn	Yêu cầu phụ thuộc
0	-	Chạy ví dụ trong đề	-
1	30	$n \leq 500$	0
2	20	$n \leq 2000$	0, 1

Phần	Điểm	Giới hạn	Yêu cầu phụ thuộc
3	20	$a_i \leq a_{i+1}$ với mọi $i < n$ (dãy không giảm)	-
4	30	Không giới hạn	0 – 3