

## BÀI TẬP LẬP TRÌNH

### THỪA SỐ NGUYÊN TỐ PHÂN BIỆT

Cho dãy số nguyên  $a_1, a_2, \dots, a_n$  và số nguyên dương  $k$ . Hãy chỉ ra các phần tử có số lượng thừa số nguyên tố phân biệt trong phân tích đúng bằng  $k$ . Chẳng hạn  $20 = 2 \times 2 \times 5$  có 2 thừa số nguyên tố phân biệt là 2 và 5.

**Dữ liệu:** vào từ tập tin văn bản **DISTINCT.INP**

- Dòng đầu tiên chứa 2 số nguyên dương  $n, k (n, k \leq 10^6)$
- Dòng tiếp theo chứa dãy  $a_1, a_2, \dots, a_n (2 \leq a_i \leq 10^6)$

**Kết quả:** ghi ra tập tin văn bản **DISTINCT.OUT** danh sách vị trí các phần tử thỏa yêu cầu đề bài theo thứ tự xuất hiện của chúng trên cùng dòng.

**Ví dụ:**

DISTINCT . INP	DISTINCT . OUT
8 3 4 30 28 154 6 12 70 11	2 4 7

## NHẬN THƯỞNG

Do nhận được giải thưởng sinh viên siêu tốt, ban tổ chức thử thách Steve trong lễ trao giải bằng cách mang ra lần lượt  $n$  phần quà, đánh số từ 1 đến  $n$ . BTC cho biết phần quà thứ  $i$  có trị giá  $a_i$ . Với mỗi phần quà mang đến, Steve được quyền nhận hoặc từ chối không nhận. Phần quà nào đã từ chối nhận trước đó rồi thì không được phép chọn lại.

Ban đầu, Steve được phép chọn bất kỳ phần quà nào. BTC yêu cầu phần quà tiếp theo được chọn phải có giá trị lớn hơn giá trị của phần quà vừa chọn trước đó, nếu không thỏa thì không được phép nhận.

**Yêu cầu:** Cho biết tổng giá trị lớn nhất của các phần quà mà Steve có thể nhận được.

**Dữ liệu:** Vào từ tập tin văn bản **AWARD.INP**

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $1 \leq n \leq 10^3$ ) – số lượng phần quà.
- Dòng tiếp theo chứa dãy số nguyên  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) – giá trị của các phần quà theo thứ tự mang đến cho Steve.

**Kết quả:** Ghi ra tập tin văn bản **AWARD.OUT** một số nguyên là tổng giá trị lớn nhất của các phần quà mà Steve có thể nhận được.

**Ví dụ:**

AWARD . INP	AWARD . OUT
6 5 <u>2</u> <u>4</u> 3 <u>8</u> 8	14

## THỐNG KÊ

Cho dãy số nguyên  $A = (a_1, a_2, \dots, a_n)$ . Với mỗi phần tử  $a_i$  trong dãy, cho biết có bao nhiêu phần tử của dãy nhỏ hơn  $a_i$  và bao nhiêu phần tử của dãy lớn hơn  $a_i$ .

**Dữ liệu:** Vào từ tập tin văn bản **STAT.INP**

- Dòng 1 chứa số nguyên dương  $n \leq 10^5$ .
- Dòng 2 chứa  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$  ( $\forall i: a_i \leq 10^9$ ).

**Kết quả:** Ghi ra tập tin văn bản **STAT.OUT**  $n$  dòng, dòng thứ  $i$  chứa hai số nguyên lần lượt là số phần tử nhỏ hơn  $a_i$  và số phần tử lớn hơn  $a_i$  trong dãy  $A$ .

**Ví dụ:**

STAT . INP	STAT . OUT
6	0 5
1 3 2 3 6 4	2 2
	1 4
	2 2
	5 0
	4 1

## SẮP XẾP SÁCH

Steve có một thư viện nhỏ gồm  $n$  cuốn sách. Do phòng hẹp, sách phải một chồng cao. Sách được đánh số từ 1 đến  $n$  theo thứ tự từ điển của tên sách. Ban đầu, các cuốn sách nằm đúng vị trí của mình từ trên xuống dưới theo đúng trình tự từ 1 đến  $n$ .

Khi cần một cuốn sách nào đó, Steve dễ dàng rút ra, nhưng sau đó không thể nhét lại vào vị trí cũ mà phải đặt nó lên trên cùng. Thời gian trôi qua và các sách không còn nằm đúng vị trí ban đầu làm cho việc tìm cuốn sách cần thiết trở nên khó khăn. Steve muốn khôi phục lại vị trí ban đầu của các cuốn sách. Để làm việc đó Steve rút ra một cuốn sách, đặt nó lên trên cùng, rồi lại rút ra cuốn khác, đặt lên trên . . . cứ như thế cho đến khi sách được sắp xếp theo thứ tự từ điển từ trên xuống dưới. Ví dụ, với  $n = 4$  và trật tự các cuốn sách từ trên xuống dưới là (1, 3, 4, 2), Steve cần rút cuốn 2 đặt lên trên, sau đó rút tiếp cuốn 1 đặt lên trên, tổng cộng là 2 lần rút sách ra đặt lại.



**Yêu cầu:** Cho  $n$  và trình tự hiện tại từ trên xuống dưới của các cuốn sách. Hãy xác định số lần tối thiểu cần rút sách ra để sắp xếp lại sách theo đúng trình tự.

**Dữ liệu:** Vào từ tập tin văn bản **BOOKSORT.INP**

- Dòng đầu tiên chứa số nguyên  $n$  ( $n \leq 3 \times 10^5$ ).
- Dòng thứ  $i$  trong  $n$  dòng sau chứa một số nguyên là thứ tự của cuốn sách ở vị trí  $i$  tính từ trên xuống.

**Kết quả:** Ghi ra tập tin văn bản **BOOKSORT.OUT** một số nguyên – số lần rút sách tối thiểu.

**Ví dụ:**

BOOKSORT . INP	BOOKSORT . OUT
4 1 3 4 2	2

## CHOCOLATE

Cho một thanh chocolate gồm các ô xếp theo hàng ngang. Ở mỗi ô có thể có chứa đúng một hạt đậu hoặc không chứa hạt đậu nào. Giữa hai ô liên tiếp có một rãnh để bẻ.

Người ta muốn bẻ thanh chocolate này làm nhiều mảnh thỏa điều kiện:

- Mỗi mảnh gồm một số ô liên tiếp của thanh chocolate ban đầu (tức là chỉ được phép bẻ thanh chocolate ở rãnh giữa hai ô liên tiếp)
- Mỗi mảnh chứa đúng một hạt đậu.

**Yêu cầu:** Cho biết có bao nhiêu cách khác nhau để bẻ thanh chocolate đạt yêu cầu trên. Hai cách bẻ được gọi là khác nhau nếu có một rãnh bị bẻ trong một cách nhưng không bị bẻ trong cách còn lại

**Dữ liệu:** Vào từ file văn bản **CHOCOLATE.INP** gồm một dòng chứa không quá 100 ký tự. Mỗi ký tự mô tả một ô trên thanh chocolate liệt kê theo thứ tự từ trái qua phải. Ký tự X có nghĩa là ô đó có chứa hạt đậu và ký tự O có nghĩa là ô đó không chứa hạt đậu nào.

**Kết quả:** Ghi ra file văn bản **CHOCOLATE.OUT** một số nguyên duy nhất là số cách bẻ tìm được.

**Ví dụ:**

CHOCOLATE . INP	CHOCOLATE . OUT
XXXXXXXX	1
XOXOX	4

*Giải thích ví dụ thứ hai: 4 cách bẻ là*

```
X|O X|O X
X|O X O|X
X O|X|O X
X O|X O|X
```

## BÀI KIỂM TRA

Cuối cùng thì Steve cũng đã đậu vào đại học sau một năm miệt mài ôn luyện cho kì thi tuyển sinh đầy căng thẳng. Steve quyết định xả hơi sau kì thi nhưng không may là kì nghỉ hè không còn đủ dài để Steve thực hiện một chuyến du lịch xa.

Steve nghe các anh chị khóa trên bảo rằng sinh viên đại học không bắt buộc phải tham dự tất cả buổi học trên lớp. Do đó Steve tìm cách trốn một số buổi học để có thể được chơi lâu hơn. Tuy nhiên, các trường đại học cũng có các buổi làm bài kiểm tra môn học tương tự như ở trường phổ thông.

Trong học phần của học kỳ này, Steve có  $n$  bài kiểm tra. Các bài kiểm tra có mức độ khó khác nhau nên sẽ có mức thang điểm khác nhau. Steve không nhất thiết phải làm đủ tất cả bài kiểm tra nhưng điểm số phải đạt tối thiểu để không bị rớt học phần. Steve nhận thấy rằng cậu ta cần ít nhất tổng cộng  $T$  điểm trong tất cả bài kiểm tra để có thể qua được môn học. Steve tự tin rằng mình có thể đạt ít nhất  $T$  điểm ở tất cả bài kiểm tra và cậu còn tự tin cao hơn khi cho rằng nếu đi thi thì mình luôn đạt trọn số điểm của bài kiểm tra đó. Steve đánh giá mức độ quan trọng của các bài kiểm tra để quyết định bài kiểm tra nào cần làm, bài nào không và nhận thấy rằng có nhiều cách khác nhau để đạt được ít nhất  $T$  điểm.

**Yêu cầu:** Cho thang điểm các bài kiểm tra và điểm số  $T$ , hãy tính số cách chọn làm các bài kiểm tra để đạt ít nhất  $T$  điểm.

**Dữ liệu:** Vào từ tập tin văn bản **EXAMS.INP**

- Dòng đầu tiên chứa hai số nguyên dương  $n, T$  ( $n \leq 36$ ).
- Dòng thứ hai chứa  $n$  số nguyên dương không vượt quá  $10^{13}$  là thang điểm của các bài kiểm tra.

**Kết quả:** Ghi ra tập tin văn bản **EXAMS.OUT** một số nguyên là kết quả bài toán.

**Ví dụ:**

EXAMS . INP	EXAMS . OUT
4 6 1 2 5 4	9

## CHUỖI GEN ĐẶC TRƯNG

Tế bào của một cá thể sinh vật ngoài hành tinh mới được phát hiện gồm rất nhiều gen, mỗi gen trong chuỗi gen của tế bào đều có số lượng nào đó các nucleotide (ký hiệu là nu). Các chuyên gia thường quan tâm chuỗi gen của mỗi cá thể dưới góc độ một chuỗi số lượng tương ứng các nu (gọi tắt là chuỗi nu), do đó chuỗi sẽ như là một dãy số nguyên dương đồng thời số số hạng của dãy này sẽ được gọi là độ dài của chuỗi. Mỗi gen được xem là đặc biệt nếu số nu của nó hoặc là bình phương của một số nguyên hoặc là lập phương của một số nguyên tố.

Để nghiên cứu khả năng biến đổi gen của loài sinh vật nói trên, các nhà khoa học xem xét hai mẫu chuỗi nu của hai cá thể và quan tâm đến mức độ “giống nhau” giữa chúng theo cách tìm ra chuỗi con chỉ gồm các gen đặc biệt mà cùng xuất hiện ở cả hai chuỗi nu (mỗi chuỗi con như vậy đều được gọi là chuỗi đặc trưng chung của hai chuỗi nu). Lưu ý rằng, chuỗi con của một chuỗi nu  $X$ , là chuỗi thu được từ  $X$  bằng cách giữ nguyên tất cả hoặc loại bỏ đi một số nào đó các gen mà vẫn giữ thứ tự xuất hiện trong chuỗi  $X$ .

**Yêu cầu:** Xác định độ dài lớn nhất  $L$  của chuỗi đặc trưng chung của hai chuỗi nu cho trước.

**Dữ liệu:** Vào từ tập tin văn bản **GEN.INP**

- Dòng đầu ghi lần lượt các số hạng của chuỗi nu thứ nhất.
- Dòng tiếp theo ghi lần lượt các số hạng của chuỗi nu thứ hai.
- Tất cả các số hạng của hai chuỗi đều nguyên dương và không vượt quá  $10^{19}$ . Độ dài của mỗi chuỗi nu đều không vượt quá 1000.

**Kết quả:** Ghi ra tập tin văn bản **GEN.OUT** số nguyên dương  $L$  tìm được.

**Ví dụ:**

GEN . INP	GEN . OUT
2 9 8 4 1 27 4 6 5 6 9 1 8 2 6 27 1 4	4

## SỐ MAY MẮN

Mỗi người có một quan điểm riêng về may mắn. Trong lĩnh vực số học, Steve coi các số ở hệ 10 chỉ chứa các chữ số 4 và 7, tức là các số chỉ chứa toàn số 4, chứa toàn số 7 hoặc cả 4 lẫn 7 là những số may mắn. Các số may mắn được Steve đánh thứ tự theo giá trị tăng dần của chúng, bắt đầu từ 1. Các số may mắn đầu tiên là 4, 7, 44, 47, . . .

**Yêu cầu:** Cho số nguyên  $k (1 \leq k \leq 10^9)$ . Hãy xác định số may mắn thứ  $k$ .

**Dữ liệu:** Vào từ tập tin văn bản **LUCKY.INP** gồm một dòng chứa số nguyên  $k$ .

**Kết quả:** Ghi ra tập tin văn bản **LUCKY.OUT** số may mắn thứ  $k$ .

**Ví dụ:**

LUCKY . INP	LUCKY . OUT
3	44



**ẢNH HIỂN THỊ**

Để một hình ảnh kích thước  $w_1 \times h_1$  hiển thị vừa vặn trên một cửa sổ kích thước  $w_2 \times h_2$  thì các phần mềm xem ảnh phải điều chỉnh kích thước của hình ảnh tăng hoặc giảm so với kích thước ban đầu. Nghĩa là kích thước một cạnh của ảnh sẽ được điều chỉnh bằng đúng kích thước một cạnh của cửa sổ, kích thước của cạnh còn lại được điều chỉnh theo tỉ lệ gần với với ảnh ban đầu nhất. Ảnh hiển thị không được xoay, có kích thước lớn nhất và không được vượt quá kích thước của cửa sổ.

**Yêu cầu:** Cho kích thước của ảnh  $w_1, h_1 (1 \leq w_1, h_1 \leq 10^9)$  và kích thước của cửa sổ  $w_2, h_2 (1 \leq w_2, h_2 \leq 10^9)$ . Hãy tính kích thước của ảnh sau khi được điều chỉnh.

**Dữ liệu:** Vào từ tập tin văn bản **FITZOOM.INP** gồm 4 số nguyên dương  $w_1, h_1, w_2, h_2$

**Kết quả:** Ghi ra tập tin văn bản **FITZOOM.OUT** 2 số nguyên dương  $w, h$  là kích thước của ảnh được điều chỉnh.

**Ví dụ:**

<b>FITZOOM.INP</b>	<b>FITZOOM.OUT</b>
1280 720 640 480	640 360

## ĐẠI DIỆN

Trên trục số cho  $n$  đoạn đóng, đoạn thứ  $i$  là  $[L_i, R_i]$ . Hãy chọn ra một tập ít nhất các điểm nguyên phân biệt trên trục số thoả mãn: Mỗi đoạn trong số  $n$  đoạn đã cho phải chứa tối thiểu 2 điểm trong tập này.

**Dữ liệu:** Vào từ tập tin văn bản **REP.INP**

- Dòng 1: chứa số nguyên dương  $n \leq 10^5$
- Trong  $n$  dòng tiếp theo, dòng thứ  $i$  chứa hai số nguyên  $L_i, R_i$  ( $\forall i: -10^6 \leq L_i < R_i \leq 10^6$ )

**Kết quả:** Ghi ra tập tin văn bản **REP.OUT**

- Dòng 1: Ghi số điểm được chọn  $m$
- Dòng 2: Ghi các toạ độ (trên trục số) của  $m$  điểm được chọn

**Ví dụ:**

REP . INP	REP . OUT
3 6 10 1 6 4 9	3 4 6 9

**MARIO**

Mario đã từng là một trò chơi rất phổ biến của học sinh. Trong một màn chơi, Mario phải vượt qua lần lượt  $n$  ống khói nằm liên tiếp nhau trên hành trình về đích. Các ống khói được đánh số từ 1 đến  $n$ , ống khói thứ  $i$  có chiều cao  $h_i$  so với mặt đất, giữa các ống khói là vực thẳm.

Ban đầu, Mario có  $K$  đơn vị năng lượng và đứng trên mặt đất. Mario phải nhảy lên ống khói số 1 để bắt đầu hành trình. Nếu Mario đang đứng trên ống khói thứ  $i$  thì tiếp theo phải nhảy đến ống khói thứ  $i + 1$ . Giả sử Mario đang có  $M$  đơn vị năng lượng, sau khi nhảy đến ống khói có chiều cao  $h$  thì mức năng lượng của Mario sẽ được tăng hoặc giảm đi theo quy luật sau:

- Nếu  $h \leq M$  thì mức năng lượng của Mario được tăng thêm  $|M - h|$  đơn vị.
- Ngược lại nếu  $h > M$  thì Mario sẽ bị mất đi  $|M - h|$  đơn vị năng lượng.

Như vậy, trong quá trình nhảy, nếu mức năng lượng hiện tại của Mario nhỏ hơn lượng bị mất đi thì Mario không thể thực hiện bước nhảy đến ống khói tiếp theo, nếu nhảy sẽ bị rơi xuống vực thẳm và mất mạng.

**Yêu cầu:** Hãy tìm  $K$  nhỏ nhất để Mario có thể nhảy được đến ống khói thứ  $n$ .

**Dữ liệu:** Vào từ tập tin văn bản **MARIO.INP**

- Dòng đầu chứa số nguyên  $n (1 \leq n \leq 10^6)$
- Dòng tiếp theo chứa dãy gồm  $n$  số nguyên  $h_1, h_2, \dots, h_n (1 \leq h_i \leq 10^6)$

**Kết quả:** Ghi ra tập tin văn bản **MARIO.OUT** giá trị  $K$  nhỏ nhất thỏa yêu cầu.

**Ví dụ:**

MARIO . INP	MARIO . OUT
5 4 2 8 6 3	4

**TIỀN**

Hóa ra ai cũng cần tiền, kể cả phù thủy. Họ sử dụng các đồng vàng, bạc và đồng, gọi tương ứng là Galeon, Sikel và Knat. Một Galeon ăn 17 sikel, một sikel ăn 29 knat. Mọi giá cả nêu sau đều theo các đơn vị kể trên. Trong mỗi giá số sikel không quá 16, số knat – không quá 28.

Trước khi vào nhập học ở Hogvard Harry Potter rút ở ngân hàng Gringot một số tiền để mua một số học cụ cần thiết như đũa thần, cú, chậu thiếc, áo choàng, . . . Số tiền Harry rút ra là  $g$  Galeo,  $s$  Sikel và  $k$  Knat. Harry cần mua tất cả là  $n$  thứ. Vật thứ  $i$  có giá là  $(p_i, q_i, r_i)$ ,  $i = 1 \div n$ ,  $(0 \leq n \leq 10^5)$ .

**Yêu cầu:** Hãy xác định số tiền Harry còn lại sau khi sắm mọi thứ. Nếu không đủ tiền thì đưa ra số  $-1$ .

**Dữ liệu:** Vào từ tập tin văn bản **MONEY.INP**

- Dòng đầu tiên chứa 3 số nguyên  $g, s, k$  ( $0 \leq g \leq 10^5$ )
- Dòng thứ 2 chứa số nguyên  $n$
- Dòng thứ  $i$  trong  $n$  dòng sau chứa 3 số nguyên  $p_i, q_i, r_i$  ( $0 \leq p_i \leq 10^5$ )

**Kết quả:** Ghi ra tập tin văn bản **MONEY.OUT** trên một dòng 3 số nguyên xác định số tiền còn lại của Harry hoặc số  $-1$ .

**Ví dụ:**

<b>MONEY . INP</b>	<b>MONEY . OUT</b>
5 16 10 2 3 9 21 0 4 0	2 2 18

## TỔNG NHỎ NHẤT

Lớp Bờm có nhiều bạn lười học, trong một lần thầy giám thị đi trực bắt gặp Bờm đang chơi trò chơi đếm số bằng điện thoại. Thầy giám thị yêu cầu Bờm viết bản kiểm điểm đồng thời giao cho  $n$  chữ số và yêu cầu Bờm sử dụng hết  $n$  chữ số đó ghép thành 2 số tự nhiên (không có chữ số 0 đứng đầu) sao cho tổng của chúng là nhỏ nhất.

**Yêu cầu:** Hãy giúp Bờm tạo 2 số tự nhiên  $a, b$  (không chứa chữ số 0 đầu tiên) từ  $n$  chữ số (có giá trị từ 0 đến 9) cho trước sao cho  $a + b$  là nhỏ nhất.

**Dữ liệu:** Vào từ tập tin văn bản **DIGITSUM.INP**

- Dòng đầu chứa số nguyên  $T$  ( $T \leq 100$ ) là số bộ test.
- Mỗi dòng trong  $T$  dòng tiếp theo ghi số nguyên dương  $n$  ( $n \leq 20$ ) và  $n$  chữ số tương ứng sau đó.

**Kết quả:** Ghi ra tập tin văn bản **DIGITSUM.OUT** gồm  $T$  dòng tương ứng tổng nhỏ nhất tìm được.

**Ví dụ:**

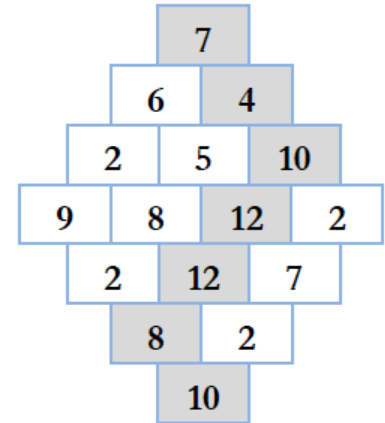
DIGITSUM . INP	DIGITSUM . OUT
2	207
5 1 2 7 8 9	447
6 3 4 2 2 2 2	

**Giải thích:**  $129 + 78 = 207$ ;  $224 + 223 = 447$

## KHI TRỘM CHUỐI

Một con khi đã tìm được cách để đột nhập vào thánh địa chuỗi gồm các lưới ô vuông bố trí theo dạng hình kim cương như minh họa. Chú khi chỉ có thể nhảy xuống các ô vuông kề cạnh bên dưới ô vuông nó đang đứng và chú khi sẽ lấy sạch lượng chuỗi ở những ô vuông mà chú đến được.

Chú khi sẽ tiến vào thánh địa chuỗi từ cửa vào là ô nằm trên cùng và sẽ thoát ra ngoài theo cửa ra là ô nằm dưới cuối của thánh địa. Chú khi muốn tìm một đường đi trong thánh địa sao cho có thể lấy được lượng chuỗi nhiều nhất có thể.



**Dữ liệu:** Vào từ tập tin văn bản **BANANAS.INP**

- Dòng đầu tiên chứa số nguyên  $T (T \leq 50)$  tương ứng số bộ test.
- Mỗi bộ test nằm trên nhiều dòng. Dòng đầu tiên chứa số nguyên  $n (1 \leq n \leq 100)$ . Tiếp theo gồm  $2 \times n - 1$  dòng mô tả số lượng chuỗi trong mỗi ô vuông tương ứng
  - + Dòng thứ  $i (1 \leq i \leq n)$  trong  $n$  dòng tiếp theo chứa  $i$  số nguyên không vượt quá  $2^{15}$
  - + Dòng thứ  $j (1 \leq j \leq n - 1)$  trong  $n - 1$  dòng tiếp theo chứa  $n - j$  số nguyên không vượt quá  $2^{15}$

**Kết quả:** Ghi ra tập tin văn bản **BANANAS.OUT** gồm  $T$  dòng, mỗi dòng tương ứng với số chuỗi thu được nhiều nhất ở từng bộ test.

**Ví dụ:**

BANANAS . INP	BANANAS . OUT
2	63
4	5
7	
6 4	
2 5 10	
9 8 12 2	
2 12 7	
8 2	
10	
2	
1	
2 3	
1	

**CỬA SỔ ỨNG DỤNG**

Khi làm việc trong môi trường Windows, ta thường mở nhiều cửa sổ ứng dụng. Mỗi cửa sổ là một hình chữ nhật có các cạnh song song với các cạnh của màn hình gồm một số ô trong lưới  $10^4 \times 10^4$  ô vuông đơn vị. Các dòng của lưới được đánh số từ 0 từ trên xuống, các cột được đánh số từ 0 từ trái sang phải. Mỗi cửa sổ được xác định bởi tọa độ ô trái trên và ô phải dưới. Nếu bấm chuột vào ô phải trên thì cửa sổ sẽ đóng lại.

Trong quá trình mở cửa sổ ứng dụng, cửa sổ mở sau có thể che một phần cửa sổ mở trước và một cửa sổ chỉ có thể đóng được nếu ô phải trên của nó không bị che.

**Yêu cầu:** Cho dãy  $n$  cửa sổ đánh số từ 1 được mở theo thứ tự đó, cần phải dùng ít nhất bao nhiêu lần đóng cửa sổ để có thể đóng được cửa sổ thứ 1.

**Dữ liệu:** Vào từ tập tin văn bản **WINS.INP**

- Dòng đầu tiên ghi số nguyên  $n \leq 1000$ .
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo ghi 4 số  $x_{i1}, y_{i1}, x_{i2}, y_{i2}$  với ý nghĩa  $(x_{i1}, y_{i1})$  là tọa độ ô trái trên và  $(x_{i2}, y_{i2})$  là tọa độ ô phải dưới của cửa sổ thứ  $i$ .

**Kết quả:** Ghi ra tập tin văn bản **WINS.OUT**

- Dòng đầu ghi số nguyên  $S$  là số lần đóng cửa sổ.
- Dòng thứ hai ghi  $S$  số hiệu của cửa sổ lần lượt đóng.

**Ví dụ:**

WINS . INP	WINS . OUT
3	2
2 3 7 7	2 1
4 1 9 5	
3 2 6 6	

## LUCKY NUMBERS

Trong một số nước châu Á, 8 và 6 được coi là những chữ số may mắn. Bất cứ số nguyên nào chỉ chứa chữ số 8 và 6 được coi là số may mắn, ví dụ 6, 8, 66, 668, 88, 886 ....

Bờm là một học sinh rất mê toán. Bờm thích các số may mắn nhưng chỉ thích các số có dạng

$$S = 8 \dots 86 \dots 6$$

trong đó  $S$  có ít nhất một chữ số 8 và chữ số 6 và 8 không nhất thiết phải đồng thời xuất hiện. Ví dụ, 8, 88, 6, 66, 86, 886, 8866 ... là các số có dạng  $S$ .

Cho trước một số nguyên dương  $X (1 < X < 10^4)$ , Nguyên muốn tìm số may mắn nhỏ nhất dạng  $S$ , có không quá 200 chữ số và chia hết cho  $X$ .

**Yêu cầu:** Hãy giúp Bờm tìm số may mắn dạng  $S$  thỏa điều kiện.

**Dữ liệu:** Vào từ tập tin văn bản **NUM86.INP** gồm nhiều bộ test

- Dòng đầu tiên chứa số nguyên dương  $T$  không lớn hơn 20 là số lượng bộ test.
- $T$  dòng tiếp theo, mỗi dòng chứa một số nguyên  $X$  tương ứng với mỗi bộ dữ liệu.

**Kết quả:** Ghi ra tập tin văn bản **NUM86.OUT** gồm  $T$  dòng, mỗi dòng ghi số may mắn dạng  $S$  nhỏ nhất chia hết cho  $X$  tương ứng với dữ liệu vào. Trường hợp không tồn tại  $S$  có không quá 200 chữ số thỏa thì ghi  $-1$ .

**Ví dụ:**

NUM86 . INP	NUM86 . OUT
4	6
6	8
8	86
43	-1
5	





## SỐ HEXA

Viết liên tiếp các số trong hệ cơ số Hexa  $1, 2, 3, \dots, 9, A, B, \dots$  ta được một dãy vô hạn các chữ số Hexa, phần đầu của dãy như sau:

123456789ABCDEF101112131415161718191A1B1C ...

**Yêu cầu:** cho số nguyên dương  $n (n \leq 10^{12})$ , tìm chữ số thứ  $n$  trong dãy trên.

**Dữ liệu:** vào từ tập tin văn bản **HEXADIGIT.INP** gồm một dòng chứa số nguyên dương  $n$ .

**Kết quả:** ghi ra tập tin văn bản **HEXADIGIT.OUT** chữ số thứ  $n$  tìm được.

**Ví dụ:**

HEXADIGIT . INP	HEXADIGIT . OUT
10	A

## TẬP SỐ

Cho số  $n$  ở hệ cơ số 10, có không quá 20 chữ số và không chứa các số 0 không có nghĩa ở đầu. Bằng cách xóa một hoặc một vài chữ số liên tiếp của  $n$  (nhưng không xóa hết tất cả các chữ số của  $n$ ) ta nhận được những số mới. Số mới được chuẩn hóa bằng cách xóa các chữ số 0 vô nghĩa nếu có. Tập số nguyên  $D$  được xây dựng bằng cách đưa vào nó số  $n$ , các số mới khác nhau đã chuẩn hóa và khác  $n$ . Ví dụ, với  $n = 1005$  ta có thể nhận được các số mới như sau:

- Bằng cách xóa một chữ số ta có các số: 5 (từ 005), 105, 105, 100;
- Bằng cách xóa hai chữ số ta có các số: 5 (từ 05), 15, 10;
- Bằng cách xóa 3 chữ số ta có các số: 5 và 1.

Tập  $D$  nhận được từ  $n$  chứa các số  $\{1005, 105, 100, 15, 10, 5, 1\}$ . Trong tập  $D$  này có 3 số chia hết cho 3, đó là các số 1005, 105 và 15.

**Yêu cầu:** Cho số nguyên  $n$ . Hãy xác định số lượng số chia hết cho 3 có mặt trong tập  $D$ .

**Dữ liệu:** Vào từ tập tin văn bản **NUMSET.INP** gồm một dòng chứa số nguyên  $n$ .

**Kết quả:** Ghi ra tập tin văn bản **NUMSET.OUT** một số nguyên – số lượng số chia hết cho 3 tìm được.

**Ví dụ:**

NUMSET . INP	NUMSET . OUT
1005	3

## CÁC SỐ HỆ ĐẾM HEXA

Cơ số của hệ đếm Hexa là 16. Để ghi các số ở hệ đếm này người ta sử dụng 16 chữ số  $0, 1, \dots, 9, A, B, C, D, E, F$ . Các chữ cái  $A..F$  được sử dụng để thay thế cho các số 10..15 tương ứng. Ví dụ số  $CF2$  trong hệ đếm Hexa chính là số  $12 \times 16^2 + 15 \times 16 + 2 = 3314$  trong hệ đếm thập phân. Giả sử  $X$  là tập các số nguyên dương có biểu diễn trong hệ đếm Hexa gồm không quá 8 chữ số và không có chữ số nào lặp lại. Ký tự trái nhất trong biểu diễn như vậy không được là 0. Số Hexa lớn nhất trong  $X$  là  $FEDCBA98$ , lớn nhì là  $FEDCBA97$ , lớn thứ ba là  $FEDCBA96$ , ...

**Yêu cầu:** Cho số nguyên dương  $n$  (không vượt quá số lượng số trong tập  $X$ ), cần tìm số lớn thứ  $n$ .

**Dữ liệu:** Vào từ tập tin văn bản **HEXA.INP** chứa số nguyên dương  $n$ .

**Kết quả:** Ghi ra tập tin văn bản **HEXA.OUT** số lớn thứ  $n$  trong tập  $X$  dưới dạng hệ đếm Hexa.

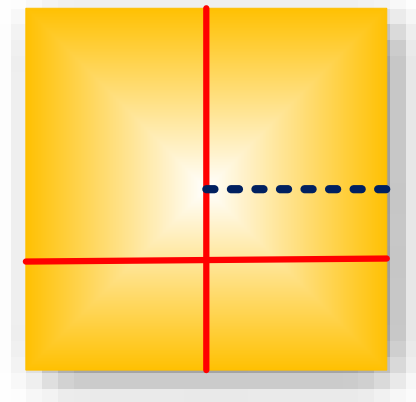
**Ví dụ:**

HEXA . INP	HEXA . OUT
11	FEDCBA87

## MIẾNG BÁNH VUÔNG

Steve được tặng một chiếc bánh ga tô sinh nhật hình chữ nhật kích thước  $m \times n$ . Steve rất thích hình vuông và dự định cắt nó ra thành các miếng hình vuông. Để làm được việc đó Steve tưởng tượng chiếc bánh như một hình chữ nhật có tọa độ góc dưới trái là  $(0, 0)$  và tọa độ góc trên phải là  $(m, n)$ .

Đang mãi suy nghĩ nên Steve không nhận thấy em mình đã nhanh tay cắt thực hiện  $k$  lần cắt, mỗi lát cắt song song với cạnh của chiếc bánh và đi suốt toàn bộ bánh. Điều này làm Steve hết sức buồn rầu chán nản và quyết định chỉ cắt một miếng hình vuông lớn nhất đặt lên bàn.



**Yêu cầu:** Cho  $n, m, k$  và các lát cắt  $t_i, v_i$ , trong đó  $t_i = 0$  xác định lát cắt  $x = v_i (0 \leq x \leq n)$ ,  $t_i = 1$  xác định lát cắt  $y = v_i (0 \leq y \leq m)$ . Hãy xác định độ dài  $d$  cạnh hình vuông lớn nhất có thể cắt.

**Dữ liệu:** vào từ tập tin văn bản **MAXPIECE.INP**

- Dòng đầu tiên chứa 3 số nguyên  $n, m, k (1 \leq n, m \leq 10^9, 0 \leq k \leq 10^5)$
- Dòng thứ  $i$  trong  $k$  dòng sau chứa 2 số nguyên  $t_i, v_i$ .

**Kết quả:** ghi ra tập tin văn bản **MAXPIECE.OUT** số nguyên  $d$ .

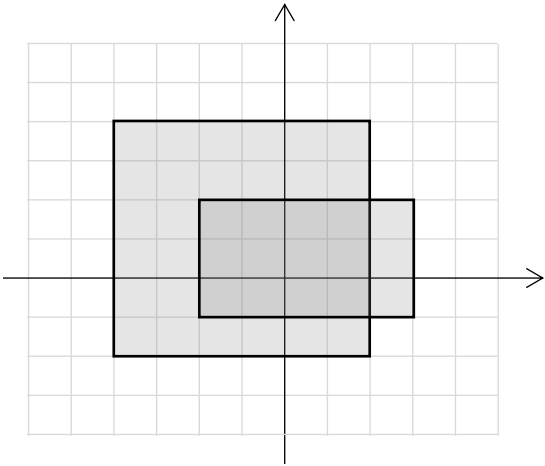
**Ví dụ:**

MAXPIECE . INP	MAXPIECE . OUT
10 10 2 1 5 0 3	5

DIỆN TÍCH PHẦN GIAO NHAU

Trong mặt phẳng, hình chữ nhật ABCD được xác định bởi tọa độ của cặp đỉnh đối nhau  $A(x_A, y_A), C(x_C, y_C)$  hoặc  $B(x_B, y_B), D(x_D, y_D)$ .

**Yêu cầu:** Cho 2 hình chữ nhật, hãy xác định diện tích phần giao nhau giữa chúng.



**Dữ liệu:** Vào từ tập tin văn bản **INTERSEC.INP**

- Dòng đầu tiên chứa số nguyên dương  $n(n \leq 10^5)$  là số lượng bộ test
- Mỗi dòng trong  $n$  dòng tiếp theo chứa 8 số nguyên  $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ . Trong đó  $x_1, y_1, x_2, y_2$  xác định hình chữ nhật thứ nhất,  $x_3, y_3, x_4, y_4$  xác định hình chữ nhật thứ hai ( $-10^8 \leq x_i, y_i \leq 10^8$ )

**Kết quả:** Ghi ra tập tin văn bản **INTERSEC.OUT** gồm  $n$  dòng là kết quả tương ứng với dữ liệu nhập.

**Ví dụ:**

INTERSEC . INP	INTERSEC . OUT
1 -4 -2 2 4 -2 2 3 -1	12

## PHÁT SINH MÃ TIẾP THEO

Một hệ thống chấm bài của một kỳ thi online tự động phát sinh mã cho thí sinh mỗi khi nhận được bài nộp lên hệ thống nếu thí sinh này trước đó chưa nộp bài ngược lại hệ thống sử dụng lại mã đã phát sinh. Mã phát sinh cho thí sinh là các chuỗi kí tự cùng độ dài  $n$  và được lấy từ tập gồm  $n$  chữ cái Latin in hoa cho trước, các kí tự trong mã thí sinh không được lặp lại.

Để tận dụng tối đa số lượng mã phát sinh và giảm thời gian tìm kiếm, hệ thống phát sinh mã cho thí sinh tăng dần theo thứ tự từ điển dựa vào thứ tự của tập chữ cái Latin in hoa cho trước. Bờm tham gia kì thi này và nhận được mã của mình. Bờm thắc mắc không biết mã của thí sinh nộp bài ngay sau mình là gì.

**Yêu cầu:** Cho tập chữ cái Latin in hoa và mã của thí sinh hiện hành. Hãy tìm mã phát sinh cho thí sinh tiếp theo hoặc ghi -1 nếu không tìm được.

**Dữ liệu:** Vào từ tập tin văn bản **NEXTCODE.INP** không quá  $10^5$  bộ test, mỗi bộ test trên 2 dòng

- Dòng đầu chứa tập chữ cái Latin in hoa độ dài  $n$  ( $n \leq 26$ ). Các kí tự viết liền nhau, không khoảng trắng và xác lập thứ tự từ điển theo thứ tự xuất hiện của các kí tự.
- Dòng thứ hai chứa chuỗi kí tự độ dài  $n$  là mã hợp lệ của thí sinh hiện hành.

**Kết quả:** Ghi ra tập tin văn bản **NEXTCODE.OUT** gồm nhiều dòng, mỗi dòng là kết quả cho từng bộ test.

**Ví dụ:**

NEXTCODE . INP	NEXTCODE . OUT
BAC ABC ABCD BADC	ACB BCAD

**TỔNG NHỎ NHẤT**

Cho hai dãy số nguyên  $A = (a_1, a_2, \dots, a_m)$  và  $B = (b_1, b_2, \dots, b_n)$  hãy tìm một phần tử  $a_i$  trong dãy  $A$  và một phần tử  $b_j$  trong dãy  $B$  có  $|a_i + b_j|$  là nhỏ nhất có thể ( $1 \leq i \leq m; 1 \leq j \leq n$ ).

**Dữ liệu:** Vào từ tập tin văn bản **ASUMMIN.INP**

- Dòng 1 chứa hai số nguyên dương  $m, n \leq 10^5$ .
- Dòng 2 chứa  $m$  số nguyên  $a_1, a_2, \dots, a_m$  ( $\forall i: |a_i| < 2^{31}$ ).
- Dòng 3 chứa  $n$  số nguyên  $b_1, b_2, \dots, b_n$  ( $\forall j: |b_j| < 2^{31}$ ).

**Kết quả:** Ghi ra tập tin văn bản **ASUMMIN.OUT** hai chỉ số  $i$  và  $j$  của hai phần tử tương ứng tìm được.

**Ví dụ**

ASUMMIN . INP	ASUMMIN . OUT
4 5 1 8 2 9 -5 -6 3 -7 -4	2 4

Giải thích:  $|a_2 + b_4| = |8 + (-7)| = 1$



## TẠO MẬT KHẨU

Quá mệt mỏi với việc thường xuyên phải nghĩ ra những dãy mật khẩu khác nhau cho hệ thống mạng của công ty nơi mà Bờm đang là nhân viên quản trị mạng. Bờm vừa nghĩ ra một cách đặt mật khẩu mới. Bờm chọn 2 từ làm cơ sở, sau đó ghép hậu tố khác rỗng của từ thứ hai vào sau tiền tố khác rỗng của từ thứ nhất để làm mật khẩu. Nói cách khác, gọi  $s_1s_2 \dots s_n$  là từ thứ nhất và  $t_1t_2 \dots t_m$  là từ thứ hai, mật khẩu tạo được có dạng  $s_1s_2 \dots s_it_jt_{j+1} \dots t_m$  ( $1 \leq i \leq n; 1 \leq j \leq m$ ).

Chẳng hạn, Bờm lấy 2 từ "cat" và "dog" làm cơ sở thì mật khẩu mới có thể như sau "cdog", "cog", "cg", "cadog", "caog", "cag", "catdog", "catog", "catg". Như vậy, Bờm có 9 mật khẩu khác nhau khi chọn "cat" và "dog" làm 2 từ cơ sở.

**Yêu cầu:** Cho 2 chuỗi có độ dài không quá  $10^5$  kí tự chữ cái in thường. Đếm số mật khẩu khác nhau có thể tạo ra được theo cách trên.

**Dữ liệu:** Vào từ tập tin văn bản **GETPASS.INP**

- Dòng thứ nhất chứa chuỗi cơ sở thứ nhất có độ dài không quá  $10^5$ .
- Dòng thứ hai chứa chuỗi cơ sở thứ hai có độ dài không quá  $10^5$ .

**Kết quả:** Ghi ra tập tin văn bản **GETPASS.OUT** một số nguyên là số mật khẩu khác nhau có thể ghép được.

**Ví dụ:**

GETPASS . INP	GETPASS . OUT
cat dog	9

## CHIA QUÀ

Bờm được tặng một miếng chocolate cực lớn hình chữ nhật kích thước  $m \times n$  được chia thành lưới ô vuông đơn vị ( $m$  hàng và  $n$  cột). Bờm muốn cắt miếng chocolate ra làm nhiều mảnh để chia cho các bạn.

Biết rằng Bờm được sử dụng không quá  $k$  nhất cắt thuộc một trong hai loại sau:

- Cắt ngang miếng chocolate từ trái qua phải theo rãnh giữa hai hàng ô liên tiếp.
- Cắt dọc miếng chocolate từ trên xuống dưới theo rãnh giữa hai cột ô liên tiếp.

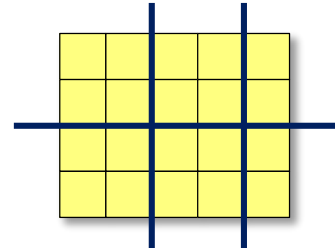
**Yêu cầu:** Giúp Bờm tìm cách cắt để chia miếng chocolate ra làm nhiều phần nhất.

**Dữ liệu:** Vào từ tập tin văn bản **CUT.INP** gồm một dòng chứa ba số nguyên dương  $m, n, k \leq 10^9$ .

**Kết quả:** Ghi ra tập tin văn bản **CUT.OUT** một số nguyên duy nhất là số phần chocolate rời nhau sau khi cắt theo phương án tìm được.

**Ví dụ:**

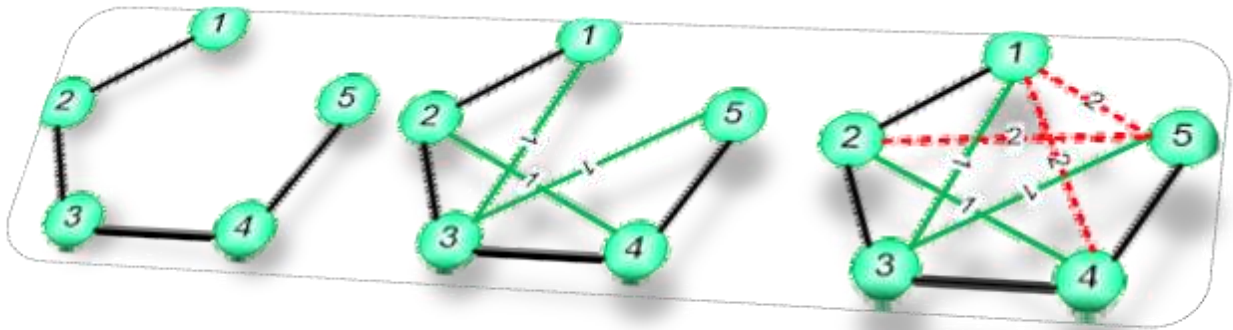
CUT . INP	CUT . OUT
4 5 3	6



## KẾT BẠN

Mạng xã hội trên Internet đã trở thành một phần của cuộc sống mỗi người. Một điều kỳ diệu thú vị là số lượng bạn bè tăng nhanh chóng một cách không ngờ. Theo quy tắc truyền thống “Bạn của bạn tôi cũng là bạn của tôi”, hàng ngày mỗi người duyệt lại danh sách những người bạn của bạn mình và bổ sung vào danh sách bạn của mình những người mới. Việc kiểm tra và xác lập quan hệ mất đúng một ngày. Nếu  $A$  và  $B$  là bạn thì  $A$  chỉ thấy được những người bạn của  $B$  có ngày hôm qua hoặc trước đó.

Các quan hệ bạn bè là đối xứng. Nếu  $A$  là bạn của  $B$  thì  $B$  cũng là bạn của  $A$ . Trong mạng xã hội mà chúng ta đang xét các mối quan hệ bạn bè luôn được duy trì giữa mọi người đã kết bạn.



**Yêu cầu:** Cho biết  $n$  – số người trong mạng,  $m$  – số mối quan hệ bạn bè đã xác lập và các mối quan hệ hiện có. Hãy xác định sau bao nhiêu ngày thì mỗi người là bạn của tất cả mọi người còn lại và mỗi ngày có bao nhiêu mối quan hệ được xác lập mới. Mỗi người được đánh số từ 1 đến  $n$ .

**Dữ liệu:** Vào từ tập tin văn bản **FRIDLIST.INP**

- Dòng đầu tiên chứa 2 số nguyên  $n, m$  ( $1 \leq n \leq 50$ ;  $1 \leq m \leq n * (n - 1) / 2$ ).
- Mỗi dòng trong  $m$  dòng sau chứa 2 số nguyên  $A$  và  $B$  cho biết giữa  $A$  và  $B$  đã có quan hệ bạn bè.

**Kết quả:** Ghi ra tập tin văn bản **FRIDLIST.OUT**

- Dòng đầu tiên đưa ra số nguyên  $k$  – số ngày xác lập quan hệ.
- Dòng thứ  $i$  trong  $k$  dòng sau chứa một số nguyên – số quan hệ được xác lập trong ngày thứ  $i$ .

**Ví dụ:**

FRIDLIST . INP	FRIDLIST . OUT
5 4	2
1 2	3
2 3	3
3 4	
4 5	

## GIÁ TRỊ CÂN BẰNG

Xét dãy số nguyên gồm  $n$  phần tử có giá trị phân biệt từ 1 đến  $n$ . Ta gọi  $x$  là giá trị cân bằng của dãy nếu số phần tử nhỏ hơn  $x$  và số phần tử lớn hơn  $x$  là bằng nhau. Như vậy chỉ có những dãy có số phần tử là lẻ thì mới có giá trị cân bằng. Chẳng hạn dãy  $\{2, 5, 8, 10, 3\}$  có giá trị cân bằng là 5.

**Yêu cầu:** Cho dãy  $a_1, a_2, \dots, a_n$  có giá trị phân biệt từ 1 đến  $n$ . Đếm số dãy con độ dài lẻ gồm các phần tử liên tiếp có giá trị cân bằng là số nguyên  $x$  cho trước.

**Dữ liệu:** Vào từ tập tin văn bản **BALANCE.INP**

- Dòng đầu tiên chứa số nguyên  $n, x (1 \leq n \leq 10^5; 1 \leq x \leq n)$
- Dòng tiếp theo chứa dãy gồm  $n$  số nguyên có giá trị phân biệt từ 1 đến  $n$ .

**Kết quả:** Ghi ra tập tin văn bản **BALANCE.OUT** một số nguyên là kết quả tìm được.

**Ví dụ:**

BALANCE . INP	BALANCE . OUT
7 4 5 7 2 4 3 1 6	4

**Giải thích:** có 4 dãy con có giá trị cân bằng là 4:  $\{4\}, \{7, 2, 4\}, \{5, 7, 2, 4, 3\}, \{5, 7, 2, 4, 3, 1, 6\}$

**DÒNG CON**

Xét xâu  $S$  độ dài  $n$ . Xâu các ký tự  $S_k, S_{k+1}, \dots, S_n, S_1, S_2, \dots, S_{k-1}$  được gọi là xâu đầy vòng của  $S$  ( $2 \leq k \leq n$ ). Bản thân  $S$  cũng được xem là xâu đầy vòng của chính nó.

Xâu con của  $S$  là xâu các ký tự  $S_i S_{i+1} \dots S_{j-1} S_j$  với cặp  $i \leq j$  nào đó.

**Yêu cầu:** Cho hai xâu  $a$  và  $b$ . Xâu  $a$  có độ dài không quá 1000, xâu  $b$  ngắn hơn hoặc bằng xâu  $a$  và có độ dài không quá 100. Các ký tự trong  $a$  và  $b$  là chữ cái la tinh (hoa hoặc thường) hay ký tự số. Hãy xác định số xâu con của  $a$  là xâu đầy vòng của  $b$ .

**Dữ liệu:** Vào từ tập tin văn bản **STRINGS.INP**:

- Dòng đầu tiên chứa xâu  $a$ ,
- Dòng thứ 2 chứa xâu  $b$ .

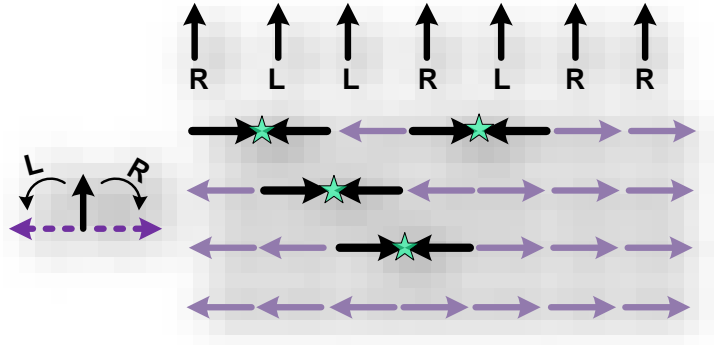
**Kết quả:** Ghi ra tập tin văn bản **STRINGS.OUT** một số nguyên.

**Ví dụ:**

STRINGS . INP	STRINGS . OUT
abcabc abc	4

## BẮT TAY

Trước buổi thi đấu các vận động viên xếp thành một hàng ngang cùng hướng mặt lên khán đài chính để chào cờ. Khi quốc ca kết thúc các vận động viên quay mặt lại bắt tay nhau. Có người quay sang phải (**R**), có người – sang trái (**L**). Nếu hai vận động viên đứng đối mặt với nhau – họ sẽ bắt tay, sau đó quay lưng lại với nhau (để có thể bắt tay với người bên cạnh). Việc bắt tay và quay lưng diễn ra đồng thời ở tất cả các cặp đối mặt nhau trong hàng và mất đúng một giây. Kết quả là có thể có các cặp vận động viên nào đó đối mặt với nhau, họ lại bắt tay và quay lưng. Quá trình trên tiếp diễn và chỉ kết thúc khi không có ai đối mặt nhau (nếu tình huống đó xuất hiện).



Ví dụ nêu ở hình bên có 7 vận động viên. Sau 3 giây quá trình bắt tay kết thúc và đã có 4 lần bắt tay.

**Yêu cầu:** Cho xâu **S** chỉ chứa các ký tự {**L**, **R**} độ dài không quá  $10^5$ . Hãy xác định quá trình bắt tay kéo dài trong bao nhiêu giây và có mấy lần bắt tay diễn ra. Nếu quá trình này không kết thúc thì đưa ra thông báo **NEVEREND**.

**Dữ liệu:** Vào từ tập tin văn bản **H\_SHAKES.INP** gồm một dòng chứa xâu **S**.

**Kết quả:** Ghi ra tập tin văn bản **H\_SHAKES.OUT** hai số nguyên kết quả hoặc thông báo

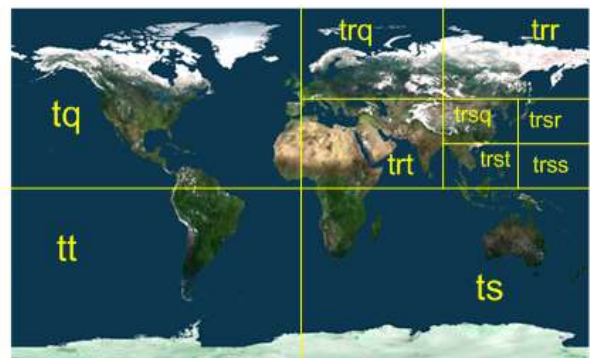
**NEVEREND**.

**Ví dụ:**

H_SHAKES . INP	H_SHAKES . OUT
RLLRLRR	3 4

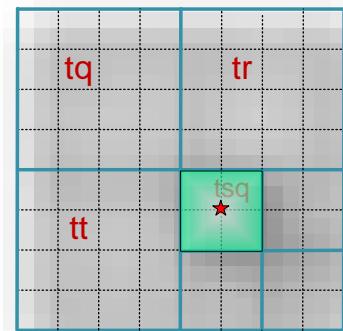
## TỌA ĐỘ VÙNG

Để mau chóng tìm kiếm được các vùng và cung cấp bản đồ tương ứng cho người dùng Google Map áp dụng cơ chế cây tứ phân. Bản đồ toàn bộ trái đất là  $t$ , được coi như một hình vuông. Bằng các đường thẳng song song với cạnh và đi qua điểm giữa của cạnh, bản đồ ban đầu được chia thành 4 phần bằng nhau, mỗi phần bằng  $1/4$  bản đồ ban đầu và ký hiệu các phần đó là  $t, s, r, q$  theo chiều ngược kim đồng hồ, bắt đầu từ phần dưới trái. Đường dẫn truy nhập tới các phần này sẽ là  $tt, ts, tr, tq$ . Mỗi phần nhận được lại có thể chia thành 4 phần con, đánh số như trên, đường dẫn tới phần con này sẽ có độ dài 3 và cứ như thế tiếp tục cho đến khi đạt được mục tiêu tìm kiếm. Ví dụ, đường dẫn tới bản đồ Việt Nam là  $trst$ .



Xét bản đồ số hình vuông tọa độ góc dưới trái là  $(0,0)$ , góc trên phải là  $(2^n, 2^n)$  với  $0 < n \leq 10000$ . Với điểm  $P$  tọa độ  $(x, y)$  ( $0 < x, y < 2^n$ ) ta có thể chỉ ra đường dẫn tới vùng nhỏ nhất chứa  $P$  như một điểm trong (không nằm trên biên). Ví dụ, với  $P$  có tọa độ  $(5,3)$  và  $n = 3$ , vùng nhỏ nhất chứa  $P$  có đường dẫn là  $tsq$ .

**Yêu cầu:** Cho  $n, x, y$ . Hãy xác định đường dẫn tới vùng chứa điểm  $(x, y)$ .



**Dữ liệu:** Vào từ tập tin văn bản **REG\_COOR.INP** gồm một dòng chứa 3 số nguyên  $n, x, y$ .

**Kết quả:** Ghi ra tập tin văn bản **REG\_COOR.OUT** xâu đường dẫn.

**Ví dụ:**

REG_COOR.INP	REG_COOR.OUT
3 5 3	tsq

## NGỌC LỤC BẢO

Ngọc lục bảo là một loại đá quý màu xanh lá cây. Đó là một loại muối của Bery có công thức hóa học là  $\text{Be}_3\text{Al}_2(\text{SiO}_3)_6$ . Những viên ngọc lục bảo xanh đậm từ 5 cara trở lên còn đắt hơn cả kim cương (một cara bằng 200mg).



Sau vài năm làm việc cật lực một nhóm 3 người gồm hai anh em và một người bạn thân quyết định chia tay nhau về nhà. Họ đã kiếm được một số ngọc đủ để về lập nghiệp với những công việc an toàn, ổn định hơn. Trước khi chia tay họ quyết định chia đều số ngọc theo khối lượng cho mỗi người: người bạn sẽ lấy một phần ba, hai anh em cùng giữ phần còn lại. Tổng cộng cả nhóm tìm được  $n$  viên ngọc, viên thứ  $i$  có khối lượng  $m_i$  cara ( $1 \leq n \leq 500; 1 \leq m_i \leq 1000$ ). Nếu có phương án chia ngọc thì họ sẽ chia ngay trước khi ra về. Trong trường hợp ngược lại – mọi người sẽ phải tìm cách bán và chia nhau bằng tiền, điều mà chưa ai muốn làm lúc này.

**Yêu cầu:** Cho  $n$  và  $m_i (i = 1 \div n)$ . Hãy xác định xem có cách chia ngọc hay không. Nếu có – chỉ ra một phương án cho biết người bạn sẽ được nhận những viên nào. Trong trường hợp không chia được – đưa ra một số 0.

**Dữ liệu:** Vào từ tập tin văn bản **EMERALD.INP**

- Dòng đầu tiên chứa số nguyên  $n$ ,
- Dòng thứ 2 chứa  $n$  số nguyên  $m_1, m_2, \dots, m_n$ .

**Kết quả:** Ghi ra tập tin văn bản **EMERALD.OUT**

- Trong trường hợp chia được:
  - o Dòng thứ nhất chứa số nguyên  $k$  – số ngọc người bạn được chia,
  - o Dòng thứ 2 chứa  $k$  số nguyên – số thứ tự các viên ngọc người bạn sẽ lấy.
- Trong trường hợp không chia được: đưa ra một số 0.

**Ví dụ:**

EMERALD . INP	EMERALD . OUT
8	2
1 3 4 1 2 5 1 1	6 8



**ƯỚC CHUNG CỦA ĐOẠN CON**

Cho dãy số nguyên  $a_1, a_2, \dots, a_n$ . Tìm một dãy con dài nhất gồm các phần tử liên tiếp thỏa: tồn tại một số nguyên  $d > 1$  sao cho mọi phần tử của dãy con này đều chia hết cho  $d$ .

**Dữ liệu:** Vào từ tập tin văn bản **CDSUBSEG.INP**

- Dòng đầu tiên chứa số nguyên  $t (t \leq 10^4)$  – số bộ test
- $t$  nhóm dòng tiếp theo, mỗi nhóm gồm 2 dòng mô tả một bộ test
  - + Dòng 1 chứa số nguyên  $n (n \leq 10^6)$ .
  - + Dòng 2 chứa dãy  $a_1, a_2, \dots, a_n (|a_i| \leq 10^6)$ .
- Tổng các giá trị  $n$  trong các bộ test không vượt quá  $10^6$ .

**Kết quả:** Ghi ra tập tin văn bản **CDSUBSEG.OUT** gồm  $t$  dòng, mỗi dòng ghi một số nguyên là độ dài dãy con tìm được của bộ test tương ứng. Nếu không tìm được dãy con thỏa điều kiện thì ghi số 0.

**Ví dụ:**

CDSUBSEG . INP	CDSUBSEG . OUT
4	1
3	4
1 2 3	6
8	1
2 6 12 15 27 1 81 5	
6	
2 4 6 8 10 12	
12	
4 5 7 9 4 5 7 9 4 5 7 9	

## CÁC VIÊN SỎI

Có  $n$  viên sỏi được xếp thành một vòng tròn, trên mỗi viên sỏi có ghi một số nguyên. Hai người chơi luân phiên lấy các viên sỏi từ vòng theo quy tắc sau:

- Người chơi đầu tiên lấy viên sỏi bất kỳ.
- Người chơi tiếp theo lấy viên sỏi nằm liền kề với viên sỏi mà người chơi trước đã lấy và lần lượt cho đến khi tất cả  $n$  viên sỏi được lấy hết.
- Khi trò chơi kết thúc thì thống kê số lượng viên sỏi mang số nguyên tố mà mỗi người lấy được. Người nào lấy được nhiều viên sỏi mang số nguyên tố hơn thì sẽ thắng ván chơi.

**Yêu cầu:** Hãy tìm chiến lược chơi để người đi đầu tiên luôn chiến thắng và đếm số lượng phương án để thắng được trò chơi.

**Dữ liệu:** Vào từ tập tin văn bản **STONES.INP**

- Dòng đầu tiên chứa số nguyên  $n (1 \leq n \leq 1000)$ .
- Dòng tiếp theo chứa dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n (|a_i| \leq 10^6)$ .

**Kết quả:** Ghi ra tập tin văn bản **STONES.OUT** một số nguyên là số phương án có thể thắng được trò chơi.

**Ví dụ:**

STONES . INP	STONES . OUT
4 5 6 7 8	2

# TỔNG CÁC CHỮ SỐ

Xét các số nguyên liên tiếp từ  $a$  đến  $b$ . Ta thực hiện lặp lại các thao tác sau:

- Bước 1: Nhân các giá trị từ  $a$  đến  $b$ .
- Bước 2: Cộng các chữ số của số nguyên nhận được.
- Bước 3: Nếu kết quả nhận được không nhỏ hơn 10 thì quay lại bước 2.

**Yêu cầu:** Tìm giá trị nhận được cuối cùng của quá trình trên.

**Dữ liệu:** Vào từ tập tin văn bản **DIGITS\_SUM.INP** chứa 2 số nguyên  $a, b (1 \leq a \leq b < 10^{100000})$ .

**Kết quả:** Ghi ra tập tin văn bản **DIGITS\_SUM.OUT** một số nguyên là kết quả của bài toán.

**Ví dụ:**

DIGITS_SUM.INP	DIGITS_SUM.OUT
1 5	3
6 8	3

**CHUỖI XOAY VÒNG**

Thao tác xoay vòng chuỗi  $s_0s_1 \dots s_{n-1}$  chuyển kí tự đầu về cuối tạo thành chuỗi  $s_1s_2 \dots s_{n-1}s_0$ . Chẳng hạn xoay vòng chuỗi "abcd" lần lượt sinh ra các chuỗi sau: "bcda", "cdab", "dabc". Xét chuỗi  $s$  chỉ gồm các kí tự chữ số từ 0 đến 9. Gọi  $S$  là tập các chuỗi xoay vòng không chứa kí tự '0' đứng đầu của  $s$ . Như vậy mỗi phần tử của  $S$  tạo thành một số thập phân không chứa số 0 vô nghĩa. Ta kí hiệu  $f(S)$  là tập các số thập phân tạo được từ tập  $S$ . Ví dụ với  $s = "2019"$  thì  $S = \{"2019", "1920", "9201"\}$  và  $f(S) = \{2019, 1920, 9201\}$ .

**Yêu cầu:** Cho 2 chuỗi  $s, t$  có độ dài không quá 3000 kí tự chỉ gồm các chữ số từ 0 đến 9. Gọi  $S, T$  là tập các chuỗi xoay vòng không chứa kí tự '0' đứng đầu của  $s, t$  tương ứng. Hãy tìm giá trị lớn nhất của  $|x - y|$ , trong đó  $x \in f(S)$  và  $y \in f(T)$ .

**Dữ liệu:** Vào từ tập tin văn bản **STRSHIFT.INP**

- Dòng đầu tiên chứa chuỗi  $s$ .
- Dòng thứ hai chứa chuỗi  $t$ .

**Kết quả:** Ghi ra tập tin văn bản **STRSHIFT.OUT** một số nguyên là giá trị lớn nhất tìm được.

**Ví dụ:**

STRSHIFT . INP	STRSHIFT . OUT
2019 2020	7181

**PHÚ ÔNG VÀ BỜM**

Một ngày rảnh rỗi, Phú Ông nghĩ ra một trò chơi gồm nhiều ván để cùng giải trí với Bờm. Mỗi ván chơi, Phú Ông đưa ra một dãy gồm  $n$  số nguyên không âm. Luật chơi như sau:

- Bờm phải chia dãy thành 2 phần khác rỗng: phần trái và phần phải sao cho tổng các phần tử của 2 phần bằng nhau. Nếu tìm được cách chia thì Bờm nhận được 1 điểm, ngược lại thì ván chơi kết thúc.
- Sau mỗi lần chia, Bờm loại bỏ phần trái hoặc phần phải ra khỏi dãy và tiếp tục trò chơi với dãy mới.

Bờm muốn đạt được nhiều điểm nhất trong trò chơi với Phú Ông. Hãy giúp Bờm tính số điểm nhiều nhất mà Bờm có thể đạt được trong mỗi ván chơi.

**Dữ liệu:** Vào từ tập tin văn bản **RELAX.INP** chứa số nguyên dương  $T (T \leq 10)$  – số ván chơi. Mỗi ván chơi được mô tả trên 2 dòng.

- Dòng đầu chứa số nguyên  $n (n \leq 10^5)$ .
- Dòng thứ hai chứa dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n (0 \leq a_i \leq 10^9)$ .

**Kết quả:** Ghi ra tập tin văn bản **RELAX.OUT** gồm  $T$  dòng, mỗi dòng ghi một số nguyên là số điểm lớn nhất mà Bờm đạt được ở mỗi ván chơi tương ứng.

**Ví dụ:**

RELAX . INP	RELAX . OUT
2	0
3	2
5 5 5	
7	
1 3 2 1 1 1 3	

## ƯỚC CHUNG LỚN NHẤT

Ước chung lớn nhất của một tập  $A$  các số nguyên dương là số nguyên dương  $d$  lớn nhất sao cho  $d$  là ước của mọi số trong tập  $A$ .

**Yêu cầu:** Cho dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$  và số  $k$  ( $2 \leq k \leq n$ ). Hãy tìm một đoạn liên tiếp các phần tử  $a_l, a_{l+1}, \dots, a_{l+k-1}$  sao cho ước chung lớn nhất của các phần tử này là lớn nhất.

**Dữ liệu:** Vào từ file văn bản **GCD.INP**

- Dòng đầu tiên chứa hai số nguyên  $n$  và  $k$  ( $2 \leq k \leq n \leq 5 \times 10^5$ ).
- Dòng thứ hai chứa  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^{18}$ ).

**Kết quả:** Ghi ra file văn bản **GCD.OUT** một số nguyên là kết quả của bài toán.

**Ví dụ:**

GCD . INP	GCD . OUT
10 3 2 6 4 3 18 12 24 8 7 5	6

**Giải thích:** Trong tất cả các bộ 3 phần tử liên tiếp của dãy (2, 6, 4, 3, 18, 12, 24, 8, 7, 5) thì ba phần tử liên tiếp (18, 12, 24) có ước chung lớn nhất bằng 6 là lớn nhất.

**MẬT ĐỘ K**

Xét 2 số nguyên  $a, b$  ( $1 \leq a \leq b \leq 10^{18}$ ) và số nguyên dương  $k$ . Người ta định nghĩa mật độ  $k$  trong đoạn  $[a, b]$  là số lượng cặp số nguyên dương  $(x, y)$  khác nhau thỏa điều kiện sau

$$\begin{cases} a \leq x^2, y^3 \leq b \\ |x^2 - y^3| \leq k \end{cases}$$

Chẳng hạn mật độ 2 của các số nguyên thuộc đoạn  $[1, 30]$  là 3 vì có 3 cặp thỏa điều kiện  $(1, 1), (3, 2), (5, 3)$ .

**Yêu cầu:** Cho 3 số nguyên  $a, b, k$  ( $1 \leq a \leq b \leq 10^{18}; 0 \leq k \leq 10^{18}$ ). Hãy tìm mật độ  $k$  trong đoạn  $[a, b]$ .

**Dữ liệu:** Vào từ tập tin văn bản **KDENSITY.INP** chứa 3 số nguyên  $a, b, k$  trên 3 dòng tương ứng.

**Kết quả:** Ghi ra tập tin văn bản **KDENSITY.OUT** một số nguyên là mật độ  $k$  trong đoạn  $[a, b]$ .

**Ví dụ:**

KDENSITY . INP	KDENSITY . OUT
1 30 2	3

## TOÁN TỬ AND

Cho dãy  $n$  số nguyên không âm  $a_1, a_2, \dots, a_n$ . Nhiệm vụ của bạn là tìm cặp hai số nguyên  $a_u, a_v$  với  $1 \leq u < v \leq n$  sao cho  $a_u \text{ AND } a_v$  là lớn nhất. Trong đó toán tử AND là phép tính trên bit.

**Dữ liệu:** Vào từ tập tin văn bản **ANDO.INP**

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $2 \leq n \leq 3 \times 10^5$ ).
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo chứa số nguyên  $a_i$  ( $0 \leq a_i \leq 10^9$ ).

**Kết quả:** Ghi ra tập tin văn bản **ANDO.OUT** một số nguyên duy nhất là kết quả bài toán.

**Ví dụ:**

ANDO . INP	ANDO . OUT
4 2 4 8 10	8



## TỔNG XOR

Cho  $n$  số nguyên dương  $a_1, a_2, \dots, a_n$ . Hãy tính giá trị biểu thức:

$$\sum_{1 \leq i < j \leq n} a_i \oplus a_j$$

Trong đó  $\oplus$  là phép cộng bit không nhớ.

**Dữ liệu:** Vào từ tập tin văn bản **SUMXOR.INP**

- Dòng đầu tiên ghi  $n$  ( $1 \leq n \leq 10^6$ ).
- $n$  dòng tiếp theo, dòng thứ  $i$  ghi số nguyên  $a_i$  ( $1 \leq a_i \leq 10^6$ ).

**Kết quả:** Ghi ra tập tin văn bản **SUMXOR.OUT** một số nguyên duy nhất là kết quả tìm được.

**Ví dụ:**

SUMXOR . INP	SUMXOR . OUT
3 7 3 5	12

## QUAN HỆ HỌ HÀNG

Để theo dõi đời sống động vật hoang dã phục vụ cho công tác bảo tồn, người ta gắn cho mỗi động vật bắt được một con chip, trong đó có chứa mã phân loại. Việc theo dõi đường di chuyển và hoạt động hàng ngày của chúng giúp cung cấp nhiều thông tin, trong đó có việc ra đời của các động vật thế hệ F2, F3, . . .

Mã phân loại là một số thập phân nguyên dương không vượt quá  $10^9$ . Các cặp mã phân loại đảm bảo nếu hai cá thể có quan hệ họ hàng với nhau thì hai mã phải có ít nhất một chữ số giống nhau. Ví dụ 2 con vật có các mã phân loại tương ứng là 47 và 107 có quan hệ họ hàng với nhau, còn hai con vật với các mã 47 và 931 – không có quan hệ.

**Yêu cầu:** Cho  $n$  mã phân loại khác nhau từng đôi một. Hãy xác định số cặp động vật có quan hệ họ hàng.

**Dữ liệu:** Vào từ tập tin văn bản **RELATIVES.INP**

- Dòng đầu tiên chứa số nguyên  $n$  ( $2 \leq n \leq 2 \times 10^5$ ).
- Dòng thứ 2 chứa  $n$  mã phân loại.

**Kết quả:** Ghi ra tập tin văn bản **RELATIVES.OUT** số cặp động vật xác định được.

**Ví dụ:**

RELATIVES . INP	RELATIVES . OUT
5 10 74 47 77 301	4

**Giải thích:** các cặp có quan hệ họ hàng: (10, 301), (74, 47), (74, 77), (47, 77).

## XÂU DÀI NHẤT

Xuất phát từ xâu  $S$  rỗng, người ta điền thêm vào cuối xâu một ký tự tùy chọn trong số  $k$  chữ cái la tinh in thường đầu tiên. Việc kéo dài xâu phải đảm bảo sao cho *không tồn tại 2 xâu con độ dài 2 ký tự liên tiếp bị lặp lại*. Người ta muốn kéo dài xâu  $S$  dài nhất có thể.

**Yêu cầu:** Cho số nguyên  $k$ . Hãy xác định xâu  $S$  dài nhất có thể nhận được. Nếu tồn tại nhiều xâu cùng có cùng độ dài lớn nhất thì đưa ra xâu bất kỳ.

**Dữ liệu:** Vào từ tập tin văn bản **STRING.INP** gồm một dòng chứa số nguyên  $k (1 \leq k \leq 26)$ .

**Kết quả:** Ghi ra tập tin văn bản **STRING.OUT** xâu tìm được.

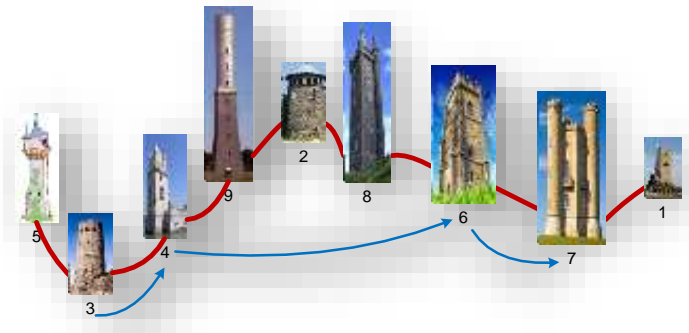
**Ví dụ:**

STRING . INP	STRING . OUT
1	aa

## TUYẾN PHÒNG VỆ

Sau cuộc chiến tranh gần nhất, đất nước bị tàn phá nặng nề. Quốc vương xứ Someland quyết định củng cố tuyến phòng thủ thủ đô. Tuyến phòng thủ bao gồm dãy các pháo đài nằm trên đường nối thủ đô với khu rừng ở phía bắc, bao gồm  $n$  pháo đài đánh số từ 1 đến  $n$  kể từ pháo đài gần thủ đô nhất, pháo đài thứ  $i$  có độ cao  $h_i$  ( $1 \leq h_i \leq 10^9$ ;  $1 \leq n \leq 2 \times 10^5$ ). Quốc vương yêu cầu các quân sư đánh giá khả năng phòng thủ. Các quân sư trình bày khá nhiều và lâu, nhưng nhà vua chỉ nắm được là chất lượng tuyến phòng thủ phụ thuộc vào dãy con các pháo đài có độ cao tăng dần tính từ thủ đô (do đặc trưng cách vây đánh của kẻ địch).

Ví dụ, với  $n = 9$  và độ cao các pháo đài là (5, 3, 4, 9, 2, 8, 6, 7, 1) thì dãy các pháo đài với độ cao (3, 4, 6, 7) sẽ quyết định chất lượng của toàn tuyến phòng thủ.



**Yêu cầu:** Cho  $n$  và các  $h_i$ . Hãy xác định số lượng pháo đài quyết định chất lượng của toàn tuyến phòng thủ.

**Dữ liệu:** Vào từ tập tin văn bản **DEFENSE.INP**

- Dòng thứ nhất chứa số nguyên  $n$ ,
- Dòng thứ hai chứa  $n$  số nguyên  $h_1, h_2, \dots, h_n$ .

**Kết quả:** Ghi ra tập tin văn bản **DEFENSE.OUT** số lượng pháo đài quyết định chất lượng của toàn tuyến phòng thủ.

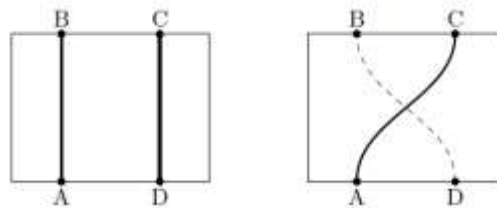
**Ví dụ:**

DEFENSE . INP	DEFENSE . OUT
9 5 3 4 9 2 8 6 7 1	4

## MẠCH IN

Mạch in được dùng trong các thiết bị điện tử là một hệ thống đường mạch được bố trí trên các phiên bản gồm một hoặc nhiều lớp. Mạch in được dùng để hàn và nối các linh kiện điện tử với nhau tạo thành bo mạch. Mạch in được làm bằng chất liệu không dẫn điện, các đường mạch trên mạch in không được chéo nhau để tránh hiện tượng đoản mạch, nếu không thì phải thiết kế các đường mạch trên các lớp khác nhau. Khi đó chi phí thiết kế bo mạch sẽ tốn kém hơn, vì vậy các nhà sản xuất tìm cách thiết kế sao cho số lớp trên phiên bản là nhỏ nhất.

Chẳng hạn xét bo mạch trong đó mỗi đường mạch nối 2 cổng trên các cạnh đối diện nhau của bo mạch như hình minh họa. Hai đường mạch nối A, B và C, D như hình bên trái thì chúng có thể được thiết kế trên cùng một lớp. Hai đường mạch nối A, C và B, D như hình bên phải được thiết kế trên các lớp khác nhau.



**Yêu cầu:** Cho vị trí các cặp đầu nối của  $n$  đường mạch trên bo mạch kích thước  $w \times h$ . Hãy xác định số lớp tối thiểu của bo mạch. Kích thước các đường mạch là không đáng kể.

**Dữ liệu:** Vào từ tập tin văn bản **BOARD.INP**

- Dòng đầu tiên chứa số nguyên  $n$  ( $1 \leq n \leq 10^5$ ) – số đường mạch trên mạch in.
- Dòng thứ  $i$  trong  $n$  dòng tiếp theo chứa hai số nguyên  $u_i, v_i$  ( $0 \leq u_i, v_i \leq 10^6$ ) mô tả đường mạch thứ  $i$  nối 2 điểm  $(u_i, 0)$  và  $(v_i, h)$ . Các điểm nối mạch phân biệt nhau.

**Kết quả:** Ghi ra tập tin văn bản **BOARD.OUT** một số nguyên là số lớp tối thiểu của bo mạch.

**Ví dụ:**

BOARD . INP	BOARD . OUT
4 0 3 1 1 3 2 2 0	3

## HÀNG RÀO

Steve phải sơn lại hàng rào cũ của nhà mình. Hàng rào bao gồm  $n$  thanh ghép khít nhau, mỗi thanh có độ rộng 1 và có độ cao có thể khác nhau, thanh thứ  $i$  có độ cao  $h_i$ . Để hoàn thành công việc nhanh chóng và dễ dàng, Steve mua một con lăn có độ rộng  $x$ . Khi sơn, con lăn phải luôn song song với mặt đất và tiếp xúc đầy đủ với hàng rào, nếu có chỗ trống, sơn sẽ bắn tung tóe làm bẩn mọi thứ xung quanh hoặc tạo ra các giọt sơn gồ ghề trên hàng rào. Vì vậy mỗi lượt, Steve phải chọn  $x$  thanh liên tiếp, đẩy con lăn từ đáy hàng rào lên đến đỉnh của thanh thấp nhất trong số các thanh đã chọn. Có thể có trường hợp một thanh được sơn tới vài ba lần, nhưng không sao, con lăn vừa phun sơn vừa xoa đều nên không để lại ảnh hưởng gì đáng kể. Steve hoàn thiện nốt phần còn lại chưa sơn của hàng rào bằng chổi sơn – một công việc nhàm chán và tốn thời gian. Vì vậy Steve phải tìm cách dùng con lăn sao cho tổng diện tích sơn bằng chổi là ít nhất.

Ví dụ, hàng rào có 5 thanh với độ cao tương ứng là 5, 3, 4, 4, 5 và con lăn có độ rộng bằng 3. Steve đẩy con lăn 2 lượt – lần đầu với các thanh 1, 2, và 3, lần thứ 2 – với các thanh 3, 4 và 5 (thanh thứ 3 được sơn 2 lần). Tổng diện tích phải sơn tay là 4.

**Yêu cầu:** Cho  $n, x$  và  $h_i$ . Hãy xác định diện tích tối thiểu phải sơn tay và số lượt đẩy con lăn.

**Dữ liệu:** Vào từ tập tin văn bản **HEDGE.INP**

- Dòng đầu tiên chứa 2 số nguyên  $n$  và  $x$  ( $1 \leq n \leq 10^6$ ;  $1 \leq x \leq n$ ).
- Dòng thứ hai chứa  $n$  số nguyên  $h_1, h_2, \dots, h_n$  ( $1 \leq h_i \leq 10^6$ ).

**Kết quả:** Ghi ra tập tin văn bản **HEDGE.OUT** gồm hai dòng

- Dòng thứ nhất ghi diện tích còn lại phải sơn bằng chổi.
- Dòng thứ hai ghi số lượt đẩy con lăn.

**Ví dụ:**

HEDGE . INP	HEDGE . OUT
5 3	3
5 3 4 4 5	2

**ĐƯỜNG CHẠY TĂNG**

Cho dãy gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n (1 \leq a_i \leq 10^9)$ . Một đoạn con gồm các phần tử liên tiếp  $a_i, a_{i+1}, \dots, a_j (1 \leq i \leq j \leq n)$  được gọi là *đường chạy tăng* nếu thỏa 3 điều kiện sau:

- $a_i \leq a_{i-1}$  hoặc  $i = 1$
- $a_j \geq a_{j+1}$  hoặc  $j = n$
- $a_i < a_{i+1} < \dots < a_j$

Có 2 loại truy vấn sau được thực hiện trên dãy:

- 1  $x$   $y$  tương ứng thao tác thay phần tử thứ  $x$  bằng giá trị  $y$ , nghĩa là  $a[x] = y$ .
- 2  $z$  tương ứng thao tác dịch xoay vòng toàn bộ dãy về bên trái  $z$  vị trí.

**Yêu cầu:** Cho dãy và các truy vấn. Đếm số lượng đường chạy tăng của dãy sau mỗi truy vấn.

**Dữ liệu:** vào từ tập tin văn bản **INCRUNS.INP**

- Dòng đầu tiên chứa số nguyên  $n (2 \leq n \leq 2 \times 10^5)$ .
- Dòng thứ hai chứa dãy  $a_1, a_2, \dots, a_n$ .
- Dòng thứ ba chứa số nguyên  $m (1 \leq m \leq 2 \times 10^5)$  tương ứng số lượng truy vấn.
- Mỗi dòng trong  $m$  dòng tiếp theo cho biết một truy vấn thuộc một trong hai dạng như mô tả.

**Kết quả:** ghi ra tập tin văn bản **INCRUNS.OUT** gồm  $m$  dòng, mỗi dòng ghi một số nguyên tương ứng với số đường chạy tăng trong dãy.

**Ví dụ:**

INCRUNS . INP	INCRUNS . OUT
9	4
3 4 4 5 7 3 2 3 10	3
4	4
1 8 9	5
1 7 5	
2 6	
1 2 3	

## SỐ KÉM MAY MẮN

Ở vương quốc Gigabyte, số 5 được coi là số kém may mắn, vì thế mọi số tự nhiên được sử dụng trong cuộc sống (dùng để đánh số nhà, làm biển số xe, giá cả các hàng hóa...) mà trong cách biểu diễn thập phân có chứa số 5 thì số đó là số kém may mắn. Ví dụ, số 123456 là số kém may mắn. Đức Vua của Gigabyte muốn biết trong đoạn  $[a; b]$  có bao nhiêu số tự nhiên kém may mắn.

**Yêu cầu:** Hãy lập trình giúp đức Vua đếm các số tự nhiên kém may mắn trong đoạn  $[a; b]$  cho trước.

**Dữ liệu:** Vào từ tập tin văn bản **UNLUCKY.INP** chứa hai số nguyên dương  $a, b (1 \leq a \leq b \leq 10^{18})$ .

**Kết quả:** Ghi ra tập tin văn bản **UNLUCKY.OUT** chứa số lượng các số tự nhiên kém may mắn thuộc  $[a; b]$ .

**Ví dụ:**

UNLUCKY . INP	UNLUCKY . OUT
1 17	2

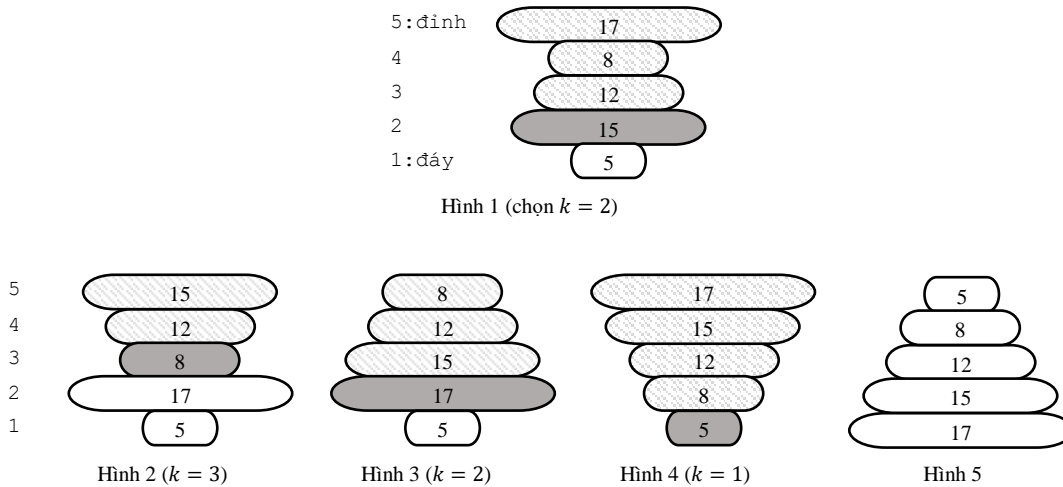


## SẮP ĐĨA

Trò chơi gồm  $n$  đĩa có đường kính phân biệt và được sắp chồng lên nhau, các đĩa được đánh số thứ tự từ 1 đến  $n$  (tính từ dưới lên như hình minh họa). Mục tiêu của trò chơi là xếp lại chồng đĩa sao cho đĩa nhỏ hơn phải nằm bên trên (có đường kính giảm dần từ dưới lên) với quy tắc như sau:

- Chọn đĩa thứ  $k$  ( $1 \leq k \leq n$ ) bất kỳ và đảo ngược chồng gồm các đĩa từ  $k$  đến  $n$ .
- Người chơi được phép thực hiện thao tác với số lần tùy ý.

Ví dụ với chồng đĩa như hình minh họa, ta lần lượt chọn các đĩa có thứ tự 2,3,2,1 và thực hiện 4 lần đảo ngược chồng đĩa.



**Yêu cầu:** Cho biết số thao tác thực hiện tối thiểu theo quy tắc trên để xếp lại chồng đĩa.

**Dữ liệu:** Vào từ tập tin văn bản **DISHES.INP**

- Dòng đầu tiên chứa số nguyên dương  $n$  ( $1 \leq n \leq 8$ ) – số lượng đĩa trong chồng.
- Trong  $n$  dòng tiếp theo, mỗi dòng chứa một số nguyên dương là đường kính của đĩa tương ứng theo đúng thứ tự trong chồng (đĩa đầu tiên có thứ tự  $n$ ) và có giá trị không vượt quá  $10^6$ .

**Kết quả:** Ghi ra tập tin văn bản **DISHES.OUT** số thao tác thực hiện tối thiểu.

**Ví dụ:**

DISHES . INP	DISHES . OUT
5 17 8 12 15 5	4

## HỆ PHƯƠNG TRÌNH LOGIC

Cho hệ phương trình sau:

$$\begin{cases} (not\ x_1\ or\ x_2\ or\ not\ x_3)\ and\ (x_1\ or\ not\ x_2\ or\ x_3) = true \\ (not\ x_2\ or\ x_3\ or\ not\ x_4)\ and\ (x_2\ or\ not\ x_3\ or\ x_4) = true \\ (not\ x_3\ or\ x_4\ or\ not\ x_5)\ and\ (x_3\ or\ not\ x_4\ or\ x_5) = true \\ \dots \\ (not\ x_k\ or\ x_{k+1}\ or\ not\ x_{k+2})\ and\ (x_k\ or\ not\ x_{k+1}\ or\ x_{k+2}) = true \end{cases}$$

trong đó  $x_i (1 \leq i \leq k + 2)$  là các biến logic (chỉ có giá trị true hoặc false).

**Yêu cầu:** Hãy đếm số bộ nghiệm của hệ phương trình trên.

**Dữ liệu:** Vào từ tập tin văn bản **BOOLSYS.INP** chứa số nguyên  $k (1 \leq k \leq 2000)$ .

**Kết quả:** Ghi ra tập tin văn bản **BOOLSYS.OUT** số bộ nghiệm của hệ phương trình.

**Ví dụ:**

BOOLSYS . INP	BOOLSYS . OUT
2	10