

BỘI SỐ CỦA K

Cho dãy n số nguyên a_1, a_2, \dots, a_n và số nguyên dương k . Hãy điền các dấu '+', '-' vào giữa các số $(a_1, a_2), (a_2, a_3), \dots, (a_{n-1}, a_n)$ để thu được biểu thức có giá trị là một số nguyên chia hết cho k .

Ví dụ: Với dãy số $A = (1, 2, 3, 4, 5)$ và $k = 3$ ta có thể có cách điền:

$$1 + 2 - 3 + 4 + 5 = 9$$

Chia hết cho $k = 3$

Dữ liệu: Vào từ file văn bản KMULT.INP

- Dòng đầu ghi hai số nguyên dương n, k ($n \leq 10^4, k \leq 100$)
- Dòng thứ hai ghi n số nguyên a_1, a_2, \dots, a_n ($|a_i| \leq 10^4$)

Kết quả: Ghi ra file văn bản KMULT.OUT

- Dòng đầu ghi 1/0 tùy theo có cách điền/không có cách điền dấu
- Nếu dòng đầu ghi 1 thì dòng thứ hai ghi $n - 1$ dấu '+' hoặc '-' thể hiện cách điền dấu hợp lệ

Example:

KMULT.INP	KMULT.OUT
5 3	1
1 2 3 4 5	---+

Ghi chú: Có 50% số test có $n \leq 20$

GIỐNG NHAU

Chúng ta định nghĩa sự giống nhau của hai xâu A và B là chiều dài của tiền tố chung dài nhất của hai xâu đó.

Ví dụ, sự giống nhau của chuỗi “abc” và “abd” là 2, còn sự giống nhau của chuỗi “aaa” và “aaab” là 3.

Bạn hãy viết chương trình tính tổng sự giống nhau của xâu S với mỗi hậu tố của nó.

Dữ liệu: Vào từ file văn bản SIMI.INP

- Gồm một dòng chứa xâu S có độ dài không vượt quá 10^6 và chỉ bao gồm các chữ cái tiếng Anh thường.

Kết quả: Ghi ra file văn bản SIMI.OUT

- Ghi ra một số nguyên là tổng sự giống nhau của xâu S với mỗi hậu tố của nó.

Ví dụ:

SIM1.INP	SIM1.OUT
ababaa	11
aa	3

Trong ví dụ đầu tiên, chuỗi $S = \text{"ababaa"}$ và các hậu tố của chuỗi S là "ababaa" , "babaa" , "abaa" , "baa" , "aa" và "a" . Sự giống nhau của mỗi chuỗi này với chuỗi S tương ứng là 6, 0, 3, 0, 1, 1. Do đó, câu trả lời là $6 + 0 + 3 + 0 + 1 + 1 = 11$.

Trong ví dụ thứ hai, câu trả lời là $2 + 1 = 3$.

Ràng buộc:

- Subtask 1 (50%): $1 \leq \text{độ dài chuỗi } S \leq 10^4$.
- Subtask 2 (50%): $10^4 < \text{độ dài chuỗi } S \leq 10^6$.

ĐÀN BÒ HỖN LOẠN

Mỗi trong N cô bò ($4 \leq N \leq 16$) của bác John có một số seri phân biệt S_i ($1 \leq S_i \leq 25.000$). Các cô bò tự hào đến nỗi mỗi cô đều đeo một chiếc vòng vàng có khắc số seri của mình trên cổ theo kiểu các băng đảng giang hồ.

Các cô bò giang hồ này thích nổi loạn nên đứng xếp hàng chờ vắt sữa theo một thứ tự gọi được gọi là 'hỗn loạn'.

Một thứ tự bò là 'hỗn loạn' nếu trong dãy số seri tạo bởi hàng bò, hai số liên tiếp khác biệt nhau nhiều hơn K ($1 \leq K \leq 3400$). Ví dụ, nếu $N = 6$ và $K = 1$ thì 1, 3, 5, 2, 6, 4 là một thứ tự 'hỗn loạn' nhưng 1, 3, 6, 5, 2, 4 thì không (vì hai số liên tiếp 5 và 6 chỉ chênh lệch 1).

Hỏi có bao nhiêu cách khác nhau để N cô bò sắp thành thứ tự 'hỗn loạn'?

Dữ liệu: Vào từ file văn bản COWMIX.INP

- Dòng 1: Hai số N và K cách nhau bởi khoảng trắng.
- Dòng $2..N + 1$: Dòng $i + 1$ chứa một số nguyên duy nhất là số seri của cô bò thứ i : S_i

Kết quả: Ghi ra file văn bản COWMIX.OUT

- Một số nguyên duy nhất là số cách để N cô bò sắp thành thứ tự 'hỗn loạn'. Kết quả đảm bảo nằm trong phạm vi kiểu số nguyên 64-bit.

Ví dụ:

COWMIX.INP	COWMIX.OUT
4 1	2
3	
4	
2	
1	

SỐ NGŨ HÀNH

Cho xâu s có n chữ số. Minh muốn xóa một vài chữ số (có thể không xóa, nhưng không được phép xóa tất cả các chữ số) để thu được “số ngũ hành”, đó là số chia hết cho 5. Lưu ý rằng, số sau khi xóa đó có thể chứa các chữ số 0 ở đầu

Bây giờ Minh muốn đếm số cách có thể thu được số ngũ hành này, số cách này chia lấy dư cho 1000000007 (tức là 10^9+7). Hai cách gọi là khác nhau nếu tập các vị trí xóa trong số đó là khác nhau.

Lưu ý rằng xâu s được cho theo một cách đặc biệt trong input

Dữ liệu: vào từ file văn bản **NGUHANH.INP**

Dòng đầu tiên cho một xâu a (chiều dài của xâu này lớn hơn hoặc bằng 1 và nhỏ hơn hoặc bằng 10^5).

Dòng thứ hai chứa một số nguyên k ($1 \leq k \leq 10^9$)

Xâu s thu được bằng cách nối k xâu a với nhau

Hay: $n = |a|k$ (chiều dài của a nhân với độ dài k)

Kết quả: ghi ra file văn bản **NGUHANH.OUT**

Đưa ra số nguyên duy nhất, số cách chia lấy dư cho $10^9 + 7$

NGUHANH.INP	NGUHANH.OUT
1256 1	4
13990 2	528
555 2	63

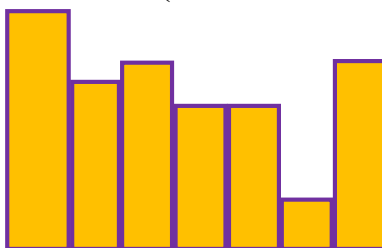
Trường hợp đầu tiên, có bốn cách để thu được số chia hết cho 5 đó là 5,15,25,và 125. Trong trường hợp thứ hai, số cần xóa là 1399013990. Trong trường hợp thứ ba, ngoại trừ xóa tất cả các chữ số, bất cứ sự lựa chọn nào cũng đúng. Sẽ có khoảng $2^6-1=63$ cách để xóa

- 30% số test với $K \leq 10^6$
- 70% số test với $K \leq 10^9$

BIỂN HIỆU (SHOPSIGN*)

Nếu các bạn biết câu chuyện thương tâm "ăn dưa leo trả vàng" của Pirate hẳn đã phải khóc hết nước mắt khi anh ấy vì lòng thương chim, đã bán rẻ trái dưa leo siêu bự của mình. Dưa leo cũng đã bị chim to lấy đi rồi, Pirate giờ chuyển sang nghề bán dưa để bù lỗ. Bất đắc dĩ thôi, vì trên đảo toàn là dưa...

Nhưng mà bán cái gì thì đầu tiên cũng phải có biển hiệu đã. Pirate quyết định lùng sục trên đảo các mảnh ván còn sót lại của những con tàu đắm để ghép lại thành tấm biển. Cuối cùng anh cũng tìm được N tấm ván hình chữ nhật, tấm thứ i có chiều rộng là 1 đơn vị và chiều dài là a_i đơn vị. Pirate dựng đứng chúng trên mặt đất và dán lại với nhau để được một mảnh ván to hơn (*xem hình minh họa*).



Việc cuối cùng chỉ là đem mảnh ván này đi ghép thành tấm biển thôi. Nhưng hóa ra đây lại là công việc khó khăn nhất. Pirate rất thích hình vuông và muốn tấm biển của mình càng to càng tốt, nhưng khổ nỗi trên đảo lại không có nhiều dụng cụ đo đạc. Không êke, không thước đo độ, nên Pirate chỉ còn cách dựa vào cạnh của N tấm ván ban đầu để ước cho bằng thôi.

Pirate chỉ có thể ước theo những đoạn thẳng chứa một cạnh nào đó (dọc hoặc ngang) của các tấm ván.

Hãy giúp anh ấy ước được tấm biển lớn nhất có thể.

Input: Vào từ file văn bản **SHOPSIGN.INP**

- Dòng thứ nhất: ghi số nguyên N – số tấm ván $1 \leq N \leq 10^6$
- N dòng tiếp theo: mô tả độ cao của các tấm ván theo thứ tự trái sang phải sau khi đã dán lại (Độ cao của các tấm ván là các số nguyên dương không vượt quá 10^9)

Output: Ghi ra file văn bản **SHOPSIGN.OUT**

- Một số nguyên duy nhất là độ dài cạnh của tấm biển lớn nhất có thể ước được.

Giới hạn:

- 60% test ứng với $1 \leq N \leq 2000$
- 20% test ứng với $N < 10^5$

Ví dụ:

SHOPSIGN.INP	SHOPSIGN.OUT
7	3
5	
2	
4	
3	
3	
1	
4	

Giải thích:

