

## A. 01. ALICEADD

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Alice có  $a$  cái kẹo, Bob cho Alice  $b$  cái kẹo, hỏi Alice có tổng cộng bao nhiêu cái kẹo. Yêu cầu viết chương trình bằng ngôn ngữ C/C++

**Lưu ý giới hạn:**  $a, b < 10^{19}$  dẫn đến  $c$  có thể vượt quá khai báo `long long`

### Input

Gồm hai dòng, mỗi dòng ghi một số nguyên

### Output

Một dòng chứa số nguyên là kết quả bài toán

### Scoring

$0 \leq a, b \leq 9 \times 10^{18}$  Có 50% test với  $a, b \leq 10^9$

### Examples

<b>input</b>	<a href="#">Copy</a>
1 1	
<b>output</b>	<a href="#">Copy</a>
2	

  

<b>input</b>	<a href="#">Copy</a>
3 5	
<b>output</b>	<a href="#">Copy</a>
8	

B. 01. SUBSEQMAX

time limit per test: 1 second  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

Tại một đại hội thể thao quốc tế có  $n$  người xếp thành hàng dài và lần lượt đi qua khán đài chào khán giả. Người thứ  $i$  giữ một mã số  $a_i$  có giá trị là một số nguyên. Alice quan sát và muốn biết một đoạn liên tiếp những người trong hàng có tổng các mã số lớn nhất là bao nhiêu.

Đề bài có thể tóm tắt như sau. Cho mảng  $s = (a_1, \dots, a_n)$ . Một đoạn  $s(i, j) = (a_i, \dots, a_j), 1 \leq i \leq j \leq n$ . Trọng số  $w(i, j) = a_i + a_{i+1} \dots + a_j$ .

Yêu cầu: Hãy tìm một đoạn trong mảng có trọng số lớn nhất, nghĩa là tổng các số trong đoạn là lớn nhất.

Input

Dòng thứ nhất chứa một số nguyên  $n \leq 10^6$  .

Dòng thứ hai chứa n số nguyên.

Output

Ghi ra duy nhất một số nguyên là trọng số lớn nhất tìm được.

Scoring

50

Example

input	Copy
6 -2 11 -4 13 -5 2	
output	Copy
20	

C. ADDMOD

time limit per test: 1 second  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

$(a+b) \bmod (10^9+7)$  a and b is in long long type

Input

contains a and b ( $1 \leq a, b \leq 18446744073709551614$ )

Output

$(a+b) \bmod (10^9+7)$

## D. EXPMOD

time limit per test: 1 second  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

Given two positive integers  $a$  and  $b$ . Compute  $a^b \bmod (10^9 + 7)$

### Input

One line contains two integers  $a$  and  $b$  ( $1 \leq a, b \leq 18446744073709551614$ )

### Output

The value  $a^b \bmod (10^9 + 7)$

### Example

input	Copy
2 3	
output	Copy
8	

## E. SUMSEQ

time limit per test: 5 seconds  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

Given a sequence  $a_1, a_2, \dots, a_n$ . Compute  $S = a_1 + a_2 + \dots + a_n$ .

$$n \leq 10^6, 0 \leq a_i \leq 10^9$$

### Input

- Line 1: number  $n$  of elements
- Line 2:  $n$  integers  $a_1, a_2, \dots, a_n$

### Output

Unique value  $S \bmod (10^9 + 7)$

### Example

input	Copy
3 18663 2391 9035	
output	Copy
30089	

## F. Way Too Long Words

time limit per test: 1 second  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

Sometimes some words like "localization" or "internationalization" are so long that writing them many times in one text is quite tiresome.

Let's consider a word *too long*, if its length is **strictly more** than 10 characters. All too long words should be replaced with a special abbreviation.

This abbreviation is made like this: we write down the first and the last letter of a word and between them we write the number of letters between the first and the last letters. That number is in decimal system and doesn't contain any leading zeroes.

Thus, "localization" will be spelt as "l10n", and "internationalization» will be spelt as "i18n".

You are suggested to automatize the process of changing the words with abbreviations. At that all too long words should be replaced by the abbreviation and the words that are not too long should not undergo any changes.

### Input

The first line contains an integer  $n$  ( $1 \leq n \leq 100$ ). Each of the following  $n$  lines contains one word. All the words consist of lowercase Latin letters and possess the lengths of from 1 to 100 characters.

### Output

Print  $n$  lines. The  $i$ -th line should contain the result of replacing of the  $i$ -th word from the input data.

### Examples

input	Copy
4 word localization internationalization pneumonoultramicroscopicsilicovolcanoconiosis	
output	Copy
word l10n i18n p43s	

## G. SMS

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Điện thoại di động (ĐTDD) trở nên một phần không thể thiếu trong cuộc sống hiện đại. Ngoài việc gọi, ĐTDD có thể gửi tin nhắn mà người ta quen gọi là SMS. Không như bàn phím máy tính, đa phần ĐTDD cổ điển hạn chế số phím. Để có thể gõ được tất cả các ký tự trong bảng chữ cái, nhiều ký tự sẽ được hiển thị trên cùng một phím. Vì vậy, để gõ một số ký tự, một phím sẽ phải được ấn liên tục đến khi ký tự cần tìm hiển thị trên màn hình.

Cho một đoạn văn bản, hãy tính số lần gõ phím để hiển thị được đoạn văn bản.

Bài toán giả thiết rằng các phím được sắp xếp như sau.

abc def

ghi jkl mno

pqrs tuv wxyz

<SP>

Trong bảng trên mỗi ô biểu diễn một phím. <SP> biểu diễn phím space. Để hiển thị ký tự 'a' thì sẽ phải bấm phím tương ứng 1 lần, nhưng để hiển thị ký tự 'b' của cùng phím đó thì sẽ phải bấm liên tục 2 lần và đối với phím 'c' là 3 lần. Tương tự, bấm 1 lần cho 'd', hai lần cho 'e' và 3 lần cho 'f'. Các ký tự khác cũng được làm tương tự. Lưu ý là để ra 1 khoảng trống thì cần bấm 1 lần phím space.

### Input

Dòng đầu tiên là một số nguyên  $T$  là số lượng trường hợp test.  $T$  dòng tiếp theo mỗi dòng chỉ chứa các khoảng trống và các ký tự in thường. Mỗi dòng chứa ít nhất 1 và tối đa 100 ký tự.

### Output

Mỗi trường hợp test đầu vào tương ứng với một dòng ở kết quả ra. Mỗi dòng bắt đầu bởi thứ tự trường hợp test và sau đó là một số biểu thị số lần bấm phím cho ra văn bản tương ứng. Xem ví dụ kết quả ra để thấy định dạng chuẩn xác.

### Example

<b>input</b>	Copy
2 welcome to ulab good luck and have fun	
<b>output</b>	Copy
Case #1: 29 Case #2: 41	

## H. SET CONTAINS BIT MASK

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Given a sequence  $a = a_1, \dots, a_n$  and a sequence  $b = b_1, \dots, b_m$ . Check if all elements of  $b$  appear in  $a$

### Input

- Line 1: contains  $n$  ( $1 \leq n \leq 31$ )
- Line 2: contains  $a_1, \dots, a_n$
- Line 3: contains  $m$  ( $1 \leq m \leq 31$ )
- Line 4: contains  $b_1, \dots, b_m$

### Output

Write 1 if all elements of  $b$  appear in  $a$

### Example

input	Copy
5 1 5 3 9 20 4 20 1 3 4	
output	Copy
0	