

Buổi 17.

Xây dựng các hàm xử lý chuỗi cơ bản.

Bài 1. Viết hàm đếm số lượng kí tự in thường trong chuỗi cho trước, chuỗi nhập vào có độ dài không vượt quá 1000 và có thể bao gồm dấu cách.

Ví dụ

Input

Ngon NGU LAP Trinh

Output

7

Bài 2. Viết hàm chuyển toàn bộ kí tự trong chuỗi cho trước về dạng in thường.

Ví dụ

Input

LAP trINH c

Output

lap trinh c

Bài 3. Viết hàm chuyển toàn bộ kí tự trong chuỗi cho trước về dạng in hoa.

Ví dụ

Input

Python java c c++

Output

PYTHON JAVA C C++

Bài 4. Viết hàm chuyển 1 kí tự từ hoa thành kí tự thường

Ví dụ

Input

ZO

Output

Z



Input

a

Output

a

Bài 5. Viết hàm chuyển 1 kí tự từ thường thành kí tự hoa.

Ví dụ

Input

a

Ouput

A

Input

ZO

utput

Z

Bài 6. Viết hàm kiểm tra xem kí tự nhập vào có phải là chữ cái. Nếu đúng trả về 1, ngược lại trả về 0.

Ví dụ

Input

A

Output

1

Bài 7. Viết hàm kiểm tra xem kí tự nhập vào có phải là chữ thường. Nếu đúng trả về 1, ngược lại trả về 0.

Ví dụ

Input

A

Ouput

0



Bài 8. Viết hàm kiểm tra xem kí tự nhập vào có phải là chữ hoa. Nếu đúng trả về 1, ngược lại trả về 0.

Ví dụ

Input

A

Output

1

Bài 9. Viết hàm kiểm tra xem kí tự nhập vào có phải là chữ số. Nếu đúng trả về 1, ngược lại trả về 0.

Ví dụ

Input

8

Output

1

Bài 10. Viết hàm kiểm tra xem kí tự nhập vào có phải chữ cái hoặc chữ số. Nếu đúng trả về 1, ngược lại trả về 0.

Input

L

Output

1

Input

#

Output

0

Bài 11. Viết hàm tính độ dài của một chuỗi.

Ví dụ

Input

Ngon ngu lap trinh



Ouput

18

Bài 12.

Hôm nay, Mezo đang chơi một trò chơi. Zoma, một nhân vật trong trò chơi đó, ban đầu ở vị trí $x = 0$. Mezo bắt đầu gửi n lệnh đến Zoma. Có hai lệnh có thể:

'L' (Trái) đặt vị trí $x: = x - 1$;

'R' (Phải) đặt vị trí $x: = x + 1$.

Thật không may, đôi khi bộ điều khiển của Mezo gặp trục trặc. Một số lệnh được gửi thành công và một số bị bỏ qua. Nếu lệnh bị bỏ qua thì vị trí x không thay đổi và Mezo chỉ cần tiến hành lệnh tiếp theo.

Ví dụ: nếu Mezo gửi lệnh "LRLR", thì đây là một số kết quả có thể xảy ra (các lệnh được gạch chân được gửi thành công):

"LRLR" - Zoma di chuyển sang trái, sang phải, sang trái một lần nữa và sang phải lần cuối cùng, kết thúc ở vị trí 0;

"LRLR" - Zoma không nhận được lệnh nào, hoàn toàn không di chuyển và kết thúc ở vị trí 0;

"LRLR" - Zoma di chuyển sang trái, sau đó sang trái một lần nữa và kết thúc ở vị trí -2 .

Mezo không biết lệnh nào sẽ được gửi thành công trước đó. Vì vậy, anh ta muốn biết Zoma có thể có bao nhiêu vị trí khác nhau.

Input

Dòng đầu tiên chứa n ($1 \leq n \leq 105$) - số lượng lệnh mà Mezo gửi.

Dòng thứ hai chứa một chuỗi s gồm n lệnh, mỗi lệnh 'L' (Trái) hoặc 'R' (Phải).

Output

In một số nguyên - số lượng vị trí khác nhau Zoma có thể kết thúc tại đó.

Ví dụ

Input

4

LRLR

Output

5

Submit tại: <https://codeforces.com/problemset/problem/1285/A>

Bài 13.

Polycarp yêu mật mã. Ông đã phát minh ra mật mã của riêng mình được gọi là repeating.

Lặp lại mật mã được sử dụng cho chuỗi. Để mã hóa chuỗi $s = s_1s_2\dots s_m$ ($1 \leq m \leq 10$), Polycarp sử dụng thuật toán sau:

anh ta viết s_1 một lần,

anh ta viết s_2 hai lần,

anh ta viết s_3 ba lần,

...

Anh ta viết xuống s_m m lần.

Ví dụ: nếu $s = \text{"bab"}$ thì quá trình là: $\text{"b"} \rightarrow \text{"baa"} \rightarrow \text{"baabbb"}$. Vì vậy, mã hóa $s = \text{"bab"}$ là "baabbb" .

Cho chuỗi t - kết quả mã hóa của một số chuỗi s . Nhiệm vụ của bạn là giải mã nó.

Input

Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 55$) - độ dài của chuỗi được mã hóa. Dòng thứ hai của đầu vào chứa t - kết quả mã hóa của một số chuỗi s . Nó chỉ chứa các chữ cái viết thường. Độ dài của t chính xác là n .

Bài toán được đảm bảo rằng câu trả lời đều tồn tại.

Output

In chuỗi ban đầu trước khi bị mã hóa.

Ví dụ

Input

6

baabbb

Output

Bab

Submit tại: <https://codeforces.com/problemset/problem/1095/A>

Bài 14.

Đôi khi một số từ như "localization" hoặc "internationalization" quá dài đến nỗi việc viết chúng nhiều lần trong một văn bản khá mệt mỏi.

Hãy xem xét một từ quá dài, nếu độ dài của nó dài hơn 10 ký tự. Tất cả các từ quá dài nên được thay thế bằng một từ viết tắt đặc biệt.

Chữ viết tắt này được thực hiện như thế này: chúng tôi viết ra chữ cái đầu tiên và chữ cái cuối cùng của một từ và giữa chúng tôi viết số lượng chữ cái giữa chữ cái đầu tiên và chữ cái cuối cùng. Con số đó nằm trong hệ thập phân và không chứa bất kỳ số 0 đứng đầu nào.

Do đó, "localization" sẽ được đánh vần là "l10n" và "internationalization" sẽ được đánh vần là "i18n".

Bạn được đề nghị tự động hóa quá trình thay đổi các từ bằng chữ viết tắt. Tại đó, tất cả các từ quá dài nên được thay thế bằng chữ viết tắt và các từ không quá dài sẽ không trải qua bất kỳ thay đổi nào.

Input

Dòng đầu tiên chứa một số nguyên n ($1 \leq n \leq 100$). Mỗi n dòng sau chứa một từ.

Tất cả các từ bao gồm các chữ cái Latinh viết thường và có độ dài từ 1 đến 100 ký tự.

Output

In n dòng. Dòng thứ i phải chứa kết quả thay thế từ thứ i từ dữ liệu đầu vào.

Ví dụ

Input

4

word

localization

internationalization

pneumonoultramicroscopicsilicovolcanoconiosis

Output

word

l10n

i18n

p43s

Submit tại: <https://codeforces.com/problemset/problem/71/A>

Bài 15.

Một ngày nọ, Vasya quyết định xem xét kết quả trận chung kết Giải vô địch bóng đá Berland 1910. Thật không may, anh ta đã không tìm thấy điểm tổng thể của trận đấu; tuy nhiên, anh ta đã nắm giữ một mô tả sâu sắc về quá trình của trận đấu. Trên tổng thể có n dòng trong mô tả đó mỗi dòng mô tả một đội ghi bàn. Giúp Vasya, tìm hiểu tên của đội đã giành chiến thắng trong trận chung kết. Nó được đảm bảo rằng trận đấu đã không kết thúc hòa.

Input

Dòng đầu tiên chứa một số nguyên n ($1 \leq n \leq 100$) - số lượng dòng trong mô tả.

Sau đó theo n dòng - cho mỗi bàn thắng tên của các đội đã ghi nó. Tên là các dòng không trống bao gồm các chữ cái Latinh viết hoa có độ dài không vượt quá 10 ký hiệu. Nó được đảm bảo rằng trận đấu không kết thúc bằng hòa và mô tả không chứa nhiều hơn hai đội khác nhau.

Output

In tên của đội chiến thắng. Chúng tôi nhắc nhở bạn rằng trong bóng đá, đội nào ghi được nhiều bàn thắng hơn được coi là người chiến thắng.

Ví dụ

Input

5

A

ABA

ABA

A

Output

A

Submit tại: <https://codeforces.com/problemset/problem/43/A>

Bài 16.

Petya bắt đầu tham dự các bài học lập trình. Trong bài học đầu tiên, nhiệm vụ của anh là viết một chương trình đơn giản. Chương trình được yêu cầu thực hiện như sau: trong chuỗi đã cho, bao gồm nếu chữ hoa và chữ thường viết hoa, nó sẽ:

xóa tất cả các nguyên âm,

chèn một ký tự "." trước mỗi phụ âm,

thay thế tất cả các phụ âm chữ hoa bằng các chữ thường tương ứng.

Nguyên âm là các chữ cái "A", "O", "Y", "E", "U", "I" và phần còn lại là phụ âm.

Đầu vào của chương trình chính xác là một chuỗi, nó sẽ trả về đầu ra dưới dạng một chuỗi, kết quả là sau khi chương trình xử lý chuỗi ban đầu.

Giúp Petya đối phó với nhiệm vụ dễ dàng này.

Input

Dòng đầu tiên đại diện cho chuỗi đầu vào của chương trình Petya. Chuỗi này chỉ bao gồm các chữ cái Latinh viết hoa và viết thường và độ dài của nó là từ 1 đến 100.

Output

In chuỗi kết quả. Nó được đảm bảo rằng chuỗi này không trống.

Ví dụ

Input

Codeforces

Output

.c.d.f.r.c.s

Submit tại : <https://codeforces.com/problemset/problem/118/A>

Bài 17.

Andrewid Android là một thám tử nổi tiếng thiên hà. Trong thời gian rảnh rỗi, anh ấy thích nghĩ về những chuỗi chứa số 0 và số 1.

Có lần anh nghĩ về một chuỗi độ dài n gồm 1 và 0. Hãy xem xét thao tác sau: chúng tôi chọn bất kỳ hai vị trí liên tiếp nào trong chuỗi và nếu một vị trí chứa 0 và vị trí kia chứa 1, thì chúng tôi được phép xóa hai chữ số này khỏi chuỗi, lấy chuỗi có độ dài $n - 2$ là một kết quả.

Bây giờ Andreid nghĩ về độ dài tối thiểu của chuỗi có thể duy trì sau khi áp dụng thao tác được mô tả nhiều lần (có thể là 0) là bao nhiêu? Giúp anh ta tính số này.

Input

Dòng đầu tiên là chiều dài n của chuỗi ($1 \leq n \leq 2 \cdot 10^5$).

Dòng thứ hai là chuỗi chỉ bao gồm 0 và 1.

Output

Chiều dài tối thiểu còn lại của dãy sau khi thực hiện các thao tác xóa như trên.

Ví dụ

Input

5

01010

Output

1

Submit tại : <https://codeforces.com/problemset/problem/556/A>

Bài 18.

Ngôn ngữ lập trình cổ điển của Bitland là Bit ++. Ngôn ngữ này rất đặc biệt và phức tạp.

Ngôn ngữ là đặc thù vì nó có chính xác một biến, được gọi là x . Ngoài ra, có hai hoạt động:

Hoạt động ++ làm tăng giá trị của biến x thêm 1.

Hoạt động - giảm giá trị của biến x xuống 1.

Một câu lệnh trong ngôn ngữ Bit ++ là một chuỗi, bao gồm chính xác một thao tác và một biến x . Câu lệnh được viết không có khoảng trắng, nghĩa là nó chỉ có thể chứa các ký tự "+", "-", "X".

Một chương trình trong Bit ++ là một chuỗi các câu lệnh, mỗi câu lệnh cần được thực thi. Thực hiện một chương trình có nghĩa là thực thi tất cả các câu lệnh mà nó chứa.

Bạn được cung cấp một chương trình bằng ngôn ngữ Bit ++. Giá trị ban đầu của x là 0. Thực hiện chương trình và tìm giá trị cuối cùng của nó (giá trị của biến khi chương trình này được thực thi).

Input

Dòng đầu tiên chứa một số nguyên n ($1 \leq n \leq 150$) - số lượng câu lệnh trong chương trình.

n dòng tiếp theo chứa một câu lệnh mỗi. Mỗi câu lệnh chứa chính xác một thao tác (++ hoặc --) và chính xác một biến x (ký hiệu là chữ «X»). Vì vậy, không có câu lệnh trống rỗng. Các hoạt động và các biến có thể được viết theo thứ tự bất kỳ.

Output

In một số nguyên duy nhất - giá trị cuối cùng của x .

Ví dụ

Input

2

X++

--X

Output

0

Submit tại : <https://codeforces.com/problemset/problem/282/A>

Bài 19.

Petya nhỏ thích quà. Mẹ anh đã mua cho anh hai chuỗi cùng kích cỡ cho ngày sinh nhật của anh. Các chuỗi bao gồm chữ hoa chữ thường và chữ thường. Bây giờ Petya muốn so sánh hai chuỗi đó theo thứ tự từ điển. Không phân biệt hoa thường, đó là một chữ cái viết hoa được coi là tương đương với chữ cái viết thường tương ứng. Giúp Petya thực hiện so sánh.

Input

Mỗi dòng trong hai dòng đầu tiên chứa một chuỗi đã mua. Độ dài của chuỗi bao gồm từ 1 đến 100. Nó được đảm bảo rằng các chuỗi có cùng độ dài và cũng bao gồm các chữ cái Latinh viết hoa và viết thường.

Output

Nếu chuỗi thứ nhất nhỏ hơn chuỗi thứ hai, hãy in "-1". Nếu chuỗi thứ hai nhỏ hơn chuỗi thứ nhất, hãy in "1". Nếu các chuỗi bằng nhau, in "0".

Ví dụ

Input

abcdefg

AbCdEfF

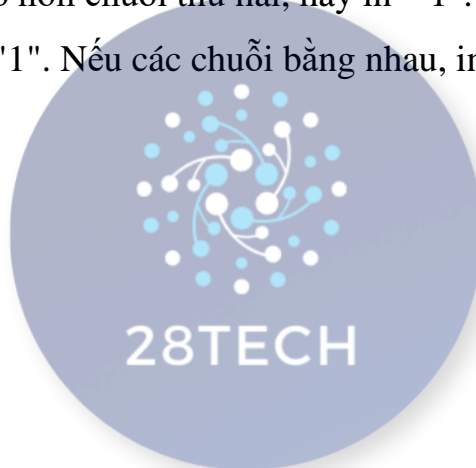
Output

1

Submit tại : <https://codeforces.com/problemset/problem/112/A>

Bài 20.

Petya rất thích bóng đá. Một ngày nọ, khi anh đang xem một trận bóng đá, anh đang viết các vị trí hiện tại của các cầu thủ trên một tờ giấy. Để đơn giản hóa tình huống, ông mô tả nó như một chuỗi bao gồm các số 0 và 1. Số 0 tương ứng với người chơi của một đội, 1 tương ứng với người chơi của một đội khác. Nếu có ít nhất 7 cầu thủ của một số đội đứng cạnh nhau, thì tình huống được coi là nguy hiểm. Ví dụ: tình huống 0010011011111101 là nguy hiểm và 11110111011101 thì không. Bạn được đưa ra tình hình hiện tại. Xác định xem nó có nguy hiểm hay không.



Input

Chuỗi bao gồm chữ số 1 và 0.

Output

Nếu có tình huống nguy hiểm, in YES, ngược lại in NO.

Ví dụ