In the store «All for O (1)» there are two cash desks and many visitors. You have to simulate the queues at these cash desks based on the recorded history of the store.

In chronological order you are given events encoded with the following characters:

- **a** – at the end of the queue, the next visitor got to the first cash desk;
- **b** – at the end of the queue, another visitor got to the second cash desk;
- **A** – at the first cash desk they served the first visitor in the queue;
- **B** – in the second cash desk they served the first visitor in line;
- **>** – The first ticket office closed;
- **]** – the second cash desk closed;
- **<** – the first cash desk opened;
- **[** – the second cash desk opened.

When the cash desk closes, all the people from the queue to this cash desk in the reverse order, starting from the last, go to the end of the other queue. That is, the person who stood last is the first to cross, then the person who stood next to last, and so on. As a result, the last in the queue will be the one who was the first in the queue to the box office that had just closed.

When a closed cash desk opens, people in line to another cashier, starting from the last, go into it if their place in the new queue is strictly less than the current one. The one who stands last becomes the first in the new line, the one who stands next to the last becomes the second and so on.

The list of events is correct, that is:

- Only closed cash desks open;
- Only open cash desks are closed;
- Visitors do not queue at closed ticket offices;
- Closed cash desks do not try to serve visitors;
- Ticket offices do not serve visitors if the lines to them are empty;
- At least one cash desk works at any given time.

Visitors are numbered from one in the order they appear in the list of events. At the initial moment, both ticket offices are open and both lines are empty.

The first line of the input contains a natural number n – the number of events ($2 \le n \le 10\ 000\ 000$). The second line contains n characters describing the events according to the above notation. It is guaranteed that the input contains at least one visitor service request.

In a single line print for each service record the last digit of the number of the served visitor. Print the answers in the order of service requests; do not use any delimiters.

Sample input 1:
15
aaabA>bBBb<BBAa

Sample output 1:
143256

Sample input 2:
12
aaaaa><AABBB

Sample output 2:
12543