Back to School '17: Big Integer

Nathan is a big fan of recreational mathematics. For one of his problems, he needs to add together very large numbers. He created a class called `BigInteger` to help with the adding, but he isn't done yet! Nathan needs to stress test his code, so he devised the following problem.

There will be N instructions (which are given as a string of length N). There are two types of instructions:

- `0` to `9`: Add this digit to the end of the current number. Afterwards, add the current number to the total.
- `-`: Remove the last digit from the current number. It is guaranteed that the current number will not be empty after this instruction. Afterwards, add the current number to the total.

At the beginning, the total is 0 and the current number is 0. Nathan wrote a program in Python to solve this problem, but it is slow and drains his battery too much. Can you help Nathan double check his answers?

Input Specification

The first line will contain the integer N.

The second line will contain a string of length N. Every character in this string can be found in `0123456789-`.

Constraints

In all subtasks, $1 \leq N \leq 500\,000$.

Subtask	Points	Additional Constraints
1	5	$N \leq 8$
2	15	After every instruction, there will be at most one non-zero digit in the current number.
3	20	The instruction `-` will not appear.
4	40	$N \leq 200\,000$
5	20	No additional constraints.

Output Specification

Print the total. Leading zeroes will be ignored by the checker. If your program does not print anything, the total is assumed to be 0.

Sample Input 1

```
8
0100---5
```

Sample Output 1

```
00000127
```

Explanation for Sample Output 1

$$00 + 001 + 0010 + 00100 + 0010 + 001 + 00 + 005 = 127$$

Notice that leading zeroes are allowed in the output.

Sample Input 2

```
4
1817
```

Sample Output 2

```
2017
```

Explanation for Sample Output 2

The numbers to be added are 01, 018, 0181, and 01817. The total is 2017.

Sample Input 3

```
2
0-
```

Sample Output 3

```
00
```