Problem A. Huge Discount

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

John heard an urban legend from Dreamoon about convenience stores in Incredible Convenient Purchasing Country (ICPC). The original price of any good there is usually as high as 10^{10^5}. Of course, nobody is going to pay such a high price. Instead, one can remove any two consecutive distinct digits from the original price. One can perform such operation as many times as he wants. Needless to say, each removal must be valid.

For example, if the original price is 123, one could pay 1 dollar by removing 23 or pay 3 dollars by removing 12. However, it's illegal to pay 2 dollars because 1 and 3 are not adjacent. However, if the original price is 111, no removal can be performed as all digits are the same.

There may be leading zeroes on the price tag. Also, leading zeroes may occur after some of such removals. In these cases, the leading zeroes are not removed automatically. Therefore, if the price tag reads 0033, one can get it for free by removing 03 twice.

John found some of such convenience stores. In these particular stores, there are some interesting properties on the prices:

1. Only digits 0, 1 and 2 are used.
2. For every i, if the first digit on the price tag of good i is removed, it becomes the price tag if good $i+1$.

For example, if the price tag of good 1 is 012, the price tag of good 2 is 12 and the price tag of good 3 is 2.

Please tell John how much it costs to buy all goods in one particular store.

Input

The first line contains an integer, n ($1 \le n \le 10^5$), the number of goods in the store.

The second line contains a string, s ($|s| = n$, $s_i \in \{0, 1, 2\}, \forall i \in [1, n]$), the price tag of good 1 in the store.

Output

An integer, the cost to buy all goods, without leading zeroes.

Examples

standard input	standard output
5 11012	3
3 111	123