G. Password

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

It's time to put your powerful eyes to use

— Authors

CPL board members created a website to let people buy final match tickets online. Jack is a massive fan of cricket. Just after he got to know about the website, he opened it in the browser, and it asked him to sign up, and now he needs to create a password.
Jack has a password generator that takes an integer n as input and displays a string s of length n consisting of lowercase Latin letters. He sets the password as string s. He is not very good at remembering passwords, so he thinks of writing them on paper, so he doesn't miss out on them.

Since it is accessible to login into his account if someone finds out about that piece of paper, he thought of another way of not missing out on his password. He comes up with a unique way of storing passwords.

Jack will do the following operation n times while maintaining a string p (which is initially empty):

In $i^{th}(1 \leq i \leq n)$ operation:

- He picks i^{th} character from string s from the left and appends it to string p.
- If i is an odd integer, he reverses the string p.

After performing n operations he writes p on the paper

Input

The first line contains a single integer $t(1 \leq t \leq 10^4)$— the number of test cases.

The first line of each test case contains a single integer $n(1 \leq n \leq 10^6)$ — the length of the string.

The second line of each test case contains a string s of length n, consisting of lowercase Latin letters.

Sum of n over all test cases does not exceed 10^6

Output

For each test case, output the string p after performing n operations.

Example

input	Copy
3 3 abc 4 efgh 5 jklmn	

output	Copy
cba gfeh nmjkl	

Note

In the first test case p which is an empty string initially becomes "a" after first operation, "ab" after second operation, "cba" after final(3^{rd}) operation

In the second test case p which is an empty string initially becomes "e" after first operation, "ef" after second operation, "gfe" after third operation and "gfeh" after final(4^{th}) operation

---