A. Even But Not Even

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

Let's define a number *ebne (even but not even)* if and only if its sum of digits is divisible by 2 but the number itself is not divisible by 2. For example, 13, 1227, 185217 are *ebne* numbers, while 12, 2, 177013, 265918 are not. If you're still unsure what *ebne* numbers are, you can look at the sample notes for more clarification.

You are given a non-negative integer s, consisting of n digits. You can delete some digits (they are **not** necessary consecutive/successive) to make the given number *ebne*. You cannot change the order of the digits, that is, after deleting the digits the remaining digits collapse. The resulting number shouldn't contain leading zeros. You can delete any number of digits between 0 (do not delete any digits at all) and $n - 1$.

For example, if you are given $s = 2223732044241852171719 12$ then one of possible ways to make it *ebne* is: ~~2~~223737~~20~~442~~4~~18521717191~~2~~ $\rightarrow$ $223734421852171719 1$. The sum of digits of $223734421852171719 1$ is equal to 70 and is divisible by 2, but number itself is not divisible by 2: it means that the resulting number is *ebne*.

Find **any** resulting number that is *ebne*. If it's impossible to create an *ebne* number from the given number report about it.

Input

The input consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 3000$) — the number of digits in the original number.

The second line of each test case contains a non-negative integer number s, consisting of n digits.

It is guaranteed that s does not contain leading zeros and the sum of n over all test cases does not exceed 3000.

Output

For each test case given in the input print the answer in the following format:

- If it is impossible to create an *ebne* number, print "`-1`" (without quotes);
- Otherwise, print the resulting number after deleting some, possibly zero, but not all digits. This number should be *ebne*. If there are multiple answers, you can print **any** of them. Note that answers with leading zeros or empty strings are not accepted. **It's not necessary to minimize or maximize the number of deleted digits**.

Example

input Copy

```
4
4
1227
1
0
6
177013
24
222373204424185217171912
```

output Copy

```
1227
-1
17703
2237344218521717191
```

Note

In the first test case of the example, 1227 is already an *ebne* number (as $1 + 2 + 2 + 7 = 12$, 12 is divisible by 2, while in the same time, 1227 is not divisible by 2) so we don't need to delete any digits. Answers such as 127 and 17 will also be accepted.

In the second test case of the example, it is clearly impossible to create an *ebne* number from the given number.

In the third test case of the example, there are many *ebne* numbers we can obtain by deleting, for example, 1 digit such as 17703, 77013 or 17013. Answers such as 1701 or 770 will not be accepted as they are not *ebne* numbers. Answer 013 will not be accepted as it contains leading zeroes.

→ Problem tags

greedy math strings *1000

No tag edit access

→ Contest materials

- Announcement (en)
- Tutorial (en)

Explanation:

- $1 + 7 + 7 + 0 + 3 = 18$. As 18 is divisible by 2 while 17703 is not divisible by 2, we can see that 17703 is an *ebne* number. Same with 77013 and 17013;
- $1 + 7 + 0 + 1 = 9$. Because 9 is not divisible by 2, 1701 is not an *ebne* number;
- $7 + 7 + 0 = 14$. This time, 14 is divisible by 2 but 770 is also divisible by 2, therefore, 770 is not an *ebne* number.

In the last test case of the example, one of many other possible answers is given. Another possible answer is: `22237320442418521717191` → `22237320442418521717191` (delete the last digit).
