Casino Floor | Advanced Division

Problem E: Slot Machines

Base Program Constraints: 1s, 256 MB

Clear Reward: +1 Point

Stanley *might* be the reason why the PHS Computer Team is now in crippling debt — but hey, it's not every day you get to visit the Cardinal Casino. Why not enjoy the various games? As long as he doesn't gamble away *too* much money, he'll be fine, and it'll make him look like a convincing customer. At least, that's what he's telling himself...

The slot machines in the casino have n **wheels**. Each wheel can display any number from 0 to 9. The state of the slot machine at any time can be represented by an string n digits long, where the i-th digit from the left represents the number on the i-th wheel. In order to win the **jackpot**, every wheel must display the **same number**. For example, 777 and 2222 are winning states, but 1234 and 0001 are not.

Luckily, Stanley has a special device with him - the **Slot Scrambler 9000**! Stanley can use this special device to modify the machine's spinning mechanisms. Let a_i represent the number on the i-th wheel. The Slot Scrambler can perform one of the two following **operations** on the machine's current state:

- Choose any $l < r$ such that $a_l = a_r$, and for every $l < i < r$, set $a_i = a_l$
- Choose any $l < r$, and replace the substring from l to r (inclusive) with the substring **sorted in ascending order**

What is the **minimum** number of times that Stanley needs to use his device to win the **jackpot** on his next pull?

Input

Each test contains multiple test cases. The first line of input contains the number of test cases t $(1 \leq t \leq 100)$.

The first line of each test case contains an integer n $(2 \leq n \leq 2 * 10^5)$, the number of wheels on the slot machine.

The next line of each test contains an n-digit string, representing the state of the slot machine.

It is guaranteed that the sum of n across test cases does not exceed $2 * 10^5$.

Output

For each test case, output one integer - the **minimum** number of device operations needed for Stanley to win the jackpot.

If it is impossible for Stanley to win using the device, output -1 instead.

Sample Test Cases

Sample 1 - Input

```
3
5
21221
7
7777777
10
0123456789
```

Sample 1 - Output

```
2
0
-1
```

Notes

In the first test case, Stanley can use one operation to sort the substring with $l = 2$ and $r = 4$, turning the state into 2112. Then, Stanley can use the first operation type to set $l = 1$ and $r = 4$, turning the state into 2222.

It can be shown that no sequence of operations can allow Stanley to win in the third test case.