

Tart Splitting

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Baker Beaver has prepared a long fruit tart. This tart is decorated with a sequence of $2N + 1$ toppings: N pieces of **I**ndigo berries, N pieces of **T**angerine slices, and exactly one **M**arzipan star. Baker Beaver will be happy if the toppings are arranged in the form $\underbrace{\text{MITIT} \dots \text{IT}}_{N \text{ times}}$. That is, the first topping should be

M, and then **I** and **T** should alternate, beginning with **I**.

However, in the morning rush, the toppings were placed in a scrambled order, represented by a string s of length $2N + 1$ consisting of the characters **M**, **I**, and **T**. To fix the tart without ruining the crust, you may perform the following operation any number of times (possibly zero):

- Choose an index i ($1 \leq i < |s|$), split the string into two parts $s[1 \dots i]$ and $s[i + 1 \dots |s|]$, and then swap the two parts to obtain the new string $s[i + 1 \dots |s|] + s[1 \dots i]$.

Find the minimum number of operations needed to make Baker Beaver happy. If it is impossible, output -1 .

Input

Each test contains multiple test cases. The first line contains a single integer T ($1 \leq T \leq 10^4$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer N ($1 \leq N \leq 10^5$).

The second line of each test case contains a string s of length $2N + 1$, consisting of N occurrences of **I**, N occurrences of **T**, and exactly one occurrence of **M**.

The sum of N across all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer — the minimum number of operations required to make Baker Beaver happy.

If it is impossible, print -1 instead.

Example

standard input	standard output
3	0
2	1
MITIT	-1
2	
TITMI	
2	
MTIIT	

Note

For the first test case, the order **MITIT** will already make Baker Beaver happy, so the answer is 0.

For the second test case, you can split the string into **TITIT** and **MI**, then swap the two parts to obtain **MITITIT**. Thus, the answer is 1.

For the third test case, it can be shown that no sequence of operations can produce **MITIT**, so the answer is -1 .