Morse (`walrus`)

In an unspecified zoo, there are N walruses[1] sitting neatly in a line. The walruses are numbered from 0 to $N-1$.

Figure 1: *Odobenus Rosmarus*, also known as *walrus*.

The state of the walruses are denoted by a string of N characters $\overline{c_0 c_1 \ldots c_{N-1}}$, where:

- $c_i =$ '`.`', if walrus i is sleeping; or
- $c_i =$ '`-`', if walrus i is awake.

In each second, the following two things happen:

1. you can choose to wake up a walrus yourself; and
2. every walrus that was awakened **in the previous second** wakes up its neighbours. The walruses which were awake at the start will **never** wake up their neighbours.

Since waking up the walruses yourself is dangerous, you ask yourself the following questions:

1. What is the smallest number of walruses you need to wake up yourself to guarantee that every walrus will be awake eventually?
2. When waking up as few walruses yourself as possible, what is the earliest possible time (in seconds) when all the walruses will be awake?

> ☞ Among the attachments of this task you may find a template file `morse.*` with a sample incomplete implementation.

[1] "*morse*" in Romanian

Input

Each test contains multiple test cases. The first line of input contains a single integer T, the number of test cases.

The first line of each test case contains a single integer N, the number of walruses.

The second line of each test case contains a string of N characters $\overline{c_0 c_1 \ldots c_{N-1}}$.

Output

For each test case, print two space separated integers in a line:

1. the minimum number of walruses you need to wake up yourself; and
2. the earliest possible time (in seconds) when all the walruses will be awake in this case.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

Constraints

- $1 \le T \le 10\,000$.
- $1 \le N \le 300\,000$.
- At least one walrus is initially asleep.
- The sum of N across all test cases does not exceed $300\,000$.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (10 points) All walruses are initially asleep.

– **Subtask 3** (20 points) $N \le 10$.

– **Subtask 4** (35 points) $T \le 10, N \le 2000$.

– **Subtask 5** (35 points) No additional constraints.

Examples

input	output
3 5 ..-.. 3 ... 20 ...---....---.--...-	2 3 1 2 4 4

Explanation

An optimal strategy for the **first test case**:

In the first second, you can wake up walrus 1 yourself:

$$``..-.." \rightarrow ``.--.."$$

In the second second, you can wake up walrus 4 yourself. Additionally, walrus 1 wakes up walrus 0:

$$``.--.." \rightarrow ``.--.-" \rightarrow ``---.-"$$

In the third second, you do not wake up any walruses yourself. However, walrus 3 is awakened by walrus 4:

$$``---.-" \rightarrow ``-----"$$

In total, you awakened 2 walruses yourself, which is the minimum required to wake up every walrus.

While waking up 2 walruses yourself, the minimum time needed to wake up every walrus is 3 seconds.

An optimal strategy for the **second test case**:

In the first second, you wake up walrus 1 yourself. The other two walruses will be woken up in the next second:

$$``..." \rightarrow ``.-." \rightarrow ``---"$$