Problem I. Move Brackets

Time limit   1000 ms
Mem limit   262144 kB

You are given a bracket sequence s of length n, where n is even (divisible by two). The string s consists of $\frac{n}{2}$ opening brackets '(' and $\frac{n}{2}$ closing brackets ')'.

In one move, you can choose **exactly one bracket** and move it to the beginning of the string or to the end of the string (i.e. you choose some index i, remove the i-th character of s and insert it before or after all remaining characters of s).

Your task is to find the minimum number of moves required to obtain **regular bracket sequence** from s. It can be proved that the answer always exists under the given constraints.

Recall what the regular bracket sequence is:

- "()" is regular bracket sequence;
- if s is regular bracket sequence then "(" + s + ")" is regular bracket sequence;
- if s and t are regular bracket sequences then $s + t$ is regular bracket sequence.

For example, "()()", "(())()", "(())" and "()" are regular bracket sequences, but ")(", "()(" and ")))" are not.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \leq t \leq 2000$) — the number of test cases. Then t test cases follow.

The first line of the test case contains one integer n ($2 \leq n \leq 50$) — the length of s. It is guaranteed that n is even. The second line of the test case containg the string s consisting of $\frac{n}{2}$ opening and $\frac{n}{2}$ closing brackets.

Output

For each test case, print the answer — the minimum number of moves required to obtain **regular bracket sequence** from s. It can be proved that the answer always exists under the

given constraints.

Examples

Input	Output
4 2)(4 ()() 8 ())()()(10)))((((())	1 0 1 3

Note

In the first test case of the example, it is sufficient to move the first bracket to the end of the string.

In the third test case of the example, it is sufficient to move the last bracket to the beginning of the string.

In the fourth test case of the example, we can choose last three openning brackets, move them to the beginning of the string and obtain "((()))(())".