C. Bad Sequence

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Petya's friends made him a birthday present — a bracket sequence. Petya was quite disappointed with his gift, because he dreamed of correct bracket sequence, yet he told his friends nothing about his dreams and decided to fix present himself.

To make everything right, Petya is going to move at most one bracket from its original place in the sequence to any other position. Reversing the bracket (e.g. turning "(" into ")" or vice versa) isn't allowed.

We remind that bracket sequence s is called correct if:

- s is empty;
- s is equal to "(t)", where t is correct bracket sequence;
- s is equal to $t_1 t_2$, i.e. concatenation of t_1 and t_2, where t_1 and t_2 are correct bracket sequences.

For example, "(()())", "()" are correct, while ")(" and "())" are not. Help Petya to fix his birthday present and understand whether he can move one bracket so that the sequence becomes correct.

Input

First of line of input contains a single number n ($1 \le n \le 200\,000$) — length of the sequence which Petya received for his birthday.

Second line of the input contains bracket sequence of length n, containing symbols "(" and ")".

Output

Print "Yes" if Petya can make his sequence correct moving at most one bracket. Otherwise print "No".

Examples

input Copy

```
2
)(
```

output Copy

```
Yes
```

input Copy

```
3
(()
```

output Copy

```
No
```

input Copy

```
2
()
```

output Copy

```
Yes
```

input Copy

```
10
))))))(((((
```

output Copy

```
No
```

Note

In the first example, Petya can move first bracket to the end, thus turning the sequence into "( )", which is correct bracket sequence.

In the second example, there is no way to move at most one bracket so that the sequence becomes correct.

In the third example, the sequence is already correct and there's no need to move brackets.