Balanced Cow Breeds

Farmer John usually brands his cows with a circular mark, but his branding iron is broken, so he instead must settle for branding each cow with a mark in the shape of a parenthesis -- (. He has two breeds of cows on his farm: Holsteins and Guernseys. He brands each of his cows with a parenthesis-shaped mark. Depending on which direction the cow is facing, this might look like either a left parenthesis or a right parenthesis.

FJ's N cows are all standing in a row, each facing an arbitrary direction, so the marks on the cows look like a string of parentheses of length N. Looking at this lineup, FJ sees a remarkable pattern: if he scans from left to right through just the Holsteins (in the order they appear in the sequence), this gives a balanced string of parentheses; moreover, the same is true for the Guernseys! To see if this is truly a rare event, please help FJ compute the number of possible ways he could assign breeds to his N cows so that this property holds.

There are several ways to define what it means for a string of parentheses to be "balanced". Perhaps the simplest definition is that there must be the same total number of ('s and)'s, and for any prefix of the string, there must be at least as many ('s as)'s. For example, the following strings are all balanced:

```
()
(())
()(()())
```

while these are not:

```
)(
())(
((()))
```

PROBLEM NAME: bbreeds

INPUT FORMAT:

* Line 1: A string of parentheses of length N (1 <= N <= 1000).

SAMPLE INPUT:

```
(())
```

OUTPUT FORMAT:

* Line 1: A single integer, specifying the number of ways FJ can assign breeds to cows so that the Holsteins form a balanced subsequence of parentheses, and likewise for the Guernseys. Since the answer might be a very large number, please print the remainder of this number when divided by 2012 (i.e., print the number mod 2012). Breed assignments involving only one breed type are valid.

SAMPLE OUTPUT:

```
6
```

OUTPUT DETAILS:

The following breed assignments work:

```
(())
HHHH

(())
GGGG

(())
HGGH

(())
GHHG

(())
HGHG

(())
GHGH
```