Boolean Parenthesization

Given a boolean expression **S** of length **N** with following symbols.
Symbols
'T' ---> true
'F' ---> false
and following operators filled between symbols
Operators
& ---> boolean AND
| ---> boolean OR
^ ---> boolean XOR
Count the number of ways we can parenthesize the expression so that the value of expression evaluates to true.

Example 1:

```
Input: N = 7
S = T|T&F^T
Output: 4
Explaination: The expression evaluates
to true in 4 ways ((T|T)&(F^T)),
(T|(T&(F^T))), (((T|T)&F)^T) and (T|((T&F)^T)).
```

Example 2:

```
Input: N = 5
S = T^F|F
Output: 2
Explaination: ((T^F)|F) and (T^(F|F)) are the
only ways.
```

Your Task:
You do not need to read input or print anything. Your task is to complete the function **countWays()** which takes N and S as input parameters and returns number of possible ways modulo 1003.

Expected Time Complexity: $O(N^3)$
Expected Auxiliary Space: $O(N^2)$

Constraints:
$1 \le N \le 200$