LEVEL-III

Parenthesization Problem

Given a boolean expression with following symbols.

Symbols
'T' ---> true
'F' ---> false

And following operators filled between symbols

Operators
& ---> boolean AND
| ---> boolean OR
^ ---> boolean XOR

Write a program to count the number of ways we can parenthesize the expression so that the value of expression evaluates to true.

Let the input be in form of two arrays one contains the symbols (T and F) in order and other contains operators (&, | and ^}

Examples:

Input: symbol[ ] = {T, F, T} operator[ ] = {^, &}

Output: 2

Explanation: The given expression is "T ^ F & T", it evaluates true
in two ways "((T ^ F) & T)" and "(T ^ (F & T))"

Input: symbol[ ] = {T, F, F} operator[ ] = {^, |}

Output: 2

Explanation: The given expression is "T ^ F | F", it evaluates true
in two ways "((T ^ F) | F)" and "(T ^ (F | F))".

Input: symbol[ ] = {T, T, F, T} operator[ ] = {|, &, ^}

Output: 4

Explanation: The given expression is "T | T & F ^ T", it evaluates true
in 4 ways ((T|T)&(F^T)), (T|(T&(F^T))), (((T|T)&F)^T) and (T|((T&F)^T)).