1106.Parsing A Boolean Expression

Return the result of evaluating a given boolean `expression`, represented as a string.

An expression can either be:

- `"t"`, evaluating to `True`;
- `"f"`, evaluating to `False`;
- `"!(expr)"`, evaluating to the logical NOT of the inner expression `expr`;
- `"&(expr1,expr2,...)"`, evaluating to the logical AND of 2 or more inner expressions `expr1, expr2, ...`;
- `"|(expr1,expr2,...)"`, evaluating to the logical OR of 2 or more inner expressions `expr1, expr2, ...`

Example 1:

```
Input: expression = "!(f)"
Output: true
```

Example 2:

```
Input: expression = "|(f,t)"
Output: true
```

Example 3:

```
Input: expression = "&(t,f)"
Output: false
```

Example 4:

```
Input: expression = "|(&(t,f,t),!(t))"
Output: false
```

Constraints:

- `1 <= expression.length <= 20000`
- `expression[i]` consists of characters in `{'(', ')', '&', '|', '!', 't', 'f', ','}`.
- `expression` is a valid expression representing a boolean, as given in the description.