

3094 Boolean Expressions

The objective of the program you are going to produce is to evaluate boolean expressions as the one shown next:

Expression: (V | V) & F & (F | V)

where 'V' is for True, and 'F' is for False. The expressions may include the following operators: '!' for **not**, '&' for **and**, '|' for **or**, the use of parenthesis for operations grouping is also allowed.

To perform the evaluation of an expression, it will be considered the priority of the operators, the not having the highest, and the or the lowest. The program must yield 'V' or 'F', as the result for each expression in the input file.

Input

The expressions are of a variable length, although will never exceed 100 symbols. Symbols may be separated by any number of spaces or no spaces at all, therefore, the total length of an expression, as a number of characters, is unknown.

The number of expressions in the input file is variable and will never be greater than 20. Each expression is presented in a new line, as shown below.

Output

For each test expression, print 'Expression ' followed by its sequence number, ': ', and the resulting value of the corresponding test expression. Separate the output for consecutive test expressions with a new line.

Use the same format as that shown in the sample output shown below.

Sample Input

```
( V | V ) & F & ( F | V )
! V | V & V & ! F & ( F | V ) & ( ! F | F | ! V & V )
( F & F | V | ! V & ! F & ! ( F | F & V ) )
```

Sample Output

```
Expression 1: F
Expression 2: V
Expression 3: V
```