A árvore binária de expressões aritméticas é uma aplicação específica de uma árvore binária para avaliar expressões aritméticas. Nesse tipo de árvore, os nós armazenam operadores binários e as folhas armazenam os operandos.

data ArvoreExpressao = No ($Int \rightarrow Int \rightarrow Int$) ArvoreExpressao ArvoreExpressao — Folha Int

Uma expressão aritmética também podem ser representada pelo seguinte tipo recursivo em Haskell:

```
data Expr = Val Int
| Soma Expr Expr
| Mult Expr Expr
| Div Expr Expr
| Sub Expr Expr
| Mod Expr Expr
deriving (Read,Eq,Show)
```

Ps: Não esqueça de colocar a Expr na classe Read

Escreva a função (eval :: $ArvoreExpressao \rightarrow Int$) tal que (eval arv) devolve o valor da expressão aritmética representada na árvore.

```
eval ( fromExpr (Soma (Val 3) (Val 4)) ) == 7
eval ( fromExpr (Mult (Val 3) (Val 4)) ) == 12
eval ( fromExpr (Div (Val 3) (Val 4)) ) == 0
eval ( fromExpr (Sub (Val 3) (Val 4)) ) == -1
```

Escreva a função (showExpr :: $Expr \rightarrow String$) tal que (showExpr expr) devolve a representação como uma string da expressão aritmética expr.

showExpr $ Mod (Mult (Val 3) (Val 4)) (Div (Val 4)(Val 3)) == "((3 * 4)%(4/3))"

showExpr $ Mod (Mult (Val 3) (Val 4)) (Div (Val 4)(Val 3)) == "((3 * 4)%(4/3))"

showExpr $ Soma (Mult (Val 3) (Val 4)) (Div (Val 4)(Val 3)) == "((3 * 4) + (4/3))"

showExpr $ Mult (Val 3) (Val 4) == "(3 * 4)"