

```
fac :: Int -> Int
```

```
fac 0 = 1
```

```
fac n = n * fac(n - 1)
```

```
mylength :: [Int] -> Int
mylength [] = 0
mylength (first:rest) = 1 + mylength rest
```

```
data Tree t
  = Nil
  | Node t (Tree t) (Tree t)
    deriving (Eq, Ord, Show)

nnodes :: Tree t -> Int
nnodes Nil = 0
nnodes (Node element left right) = 1 + nnodes(left) + nnodes(right)
```