factorial (recursive)

(Haskell) Saturday 29th October, 2016

factorial (non-recursive)

(Haskell) Saturday 29th October, 2016

(*)

(Haskell) Saturday 29th October, 2016

product

(Haskell) Saturday 29th October, 2016

length

(Haskell) Saturday 29th October, 2016

reverse

(Haskell) Saturday 29th October, 2016

```
factorial    :: Int → Int
factorial n  =  product [1..n]
```

```
factorial          :: Int → Int
factorial 0        =  1
factorial (n + 1)  =  (n + 1) * factorial n
```

```
product          :: Num a ⇒ [a] → a
product []       =  1
product (n : ns) =  n * product ns
```

```
(*)          :: Int → Int → Int
m * 0        =  0
m * (n + 1)  =  m + (m * n)
```

```
reverse          :: [a] → [a]
reverse []       =  []
reverse (x : xs) =  reverse xs ++ [x]
```

```
length          :: [a] → Int
length []       =  0
length [_ : xs] =  1 + length xs
```