Exercise 4.1
Assume the following Haskell definitions:

\[
\begin{align*}
\text{size} & = 10 \\
\text{square } n & = n \times n
\end{align*}
\]

Add a definition for \textit{cube} with the obvious meaning, and manually perform single-stepped expression evaluation for the expression “\textit{cube size} - \textit{cube (size - 2)}”.

Exercise 4.2
Haskell has predefined types \textit{Float} for single-precision floating point numbers (which we ignore in the following) and \textit{Double} for double-precision floating point numbers.

Standard mathematical functions like

\[
\begin{align*}
\text{sqrt}, \ \text{sin}, \ \text{atan} & : \text{Double} \rightarrow \text{Double}
\end{align*}
\]

and \textit{pi} : \textit{Double} are also available; \textit{x} \wedge k stands for \textit{x}^k if \textit{k} is natural; \textit{x} \times q can be used for \textit{x}^q where both \textit{x} and \textit{q} are of type \textit{Double}.

Define the following Haskell functions, with the meanings obvious from their names:

(a) \textit{sphereVolume} :: \text{Double} \rightarrow \text{Double}
(b) \textit{sphereSurface} :: \text{Double} \rightarrow \text{Double}
(c) \textit{centuryToPicosecond} :: \text{Integer} \rightarrow \text{Integer}

Try the last one in C or Java, too; test both, and compare the results

Exercise 4.3
Define the following Haskell functions:

(a) \textit{stutter} :: [a] \rightarrow [a]

duplicates each element of its argument lists, e.g.: \textit{stutter} [1,2,3] = [1,1,2,2,3,3]

(b) \textit{splits} :: [a] \rightarrow [[[a],[a]]]

delivers for each argument list all possibilities to segment it into non-empty prefix and suffix, e.g.:
\textit{splits} [1,2,3] = [[[1],[2,3]],([[1],[2]],[[1],[3]])]
(The order is irrelevant.)

(c) \textit{rotations} :: [a] \rightarrow [[[a]]]

delivers for each argument list all different results of rotations, each result only once, e.g.:
rotations \([1,2,3] = [[[1,2,3], [3,1,2], [2,3,1]]
\)
(The order is irrelevant.)

(d) \(\textit{permutations} :: [a] \rightarrow [[[a]]]
\)

delivers for each argument list all different results of permutations, each result only once, e.g.:
\(\textit{permutations} \{1,2,3\} = [[[1,2,3], [1,3,2], [2,1,3], [2,3,1], [3,1,2], [3,2,1]]
\)
(The order is irrelevant.)

\textbf{Exercise 4.4 — Defining Haskell Functions (40\% of Midterm 1, 2003)}

Define the following Haskell functions (the solutions are independent of each other):

(a) \(\textit{polynomial} :: [\text{Double}] \rightarrow \text{Double} \rightarrow \text{Double}
\)
such that for coefficients \(c_0, c_1, c_2, \ldots, c_n\) and any \(x\) the following holds:
\[
\textit{polynomial} [c_0, c_1, c_2, \ldots, c_n] x = c_0 + c_1 \cdot x + c_2 \cdot x^2 + \cdots + c_n \cdot x^n
\]
e.g.: \(\textit{polynomial} [3,4,5] 100.0 = 50403.0
\)

\textbf{Hint:} Use Horner's rule:
\[
c_0 + c_1 \cdot x + c_2 \cdot x^2 + \cdots + c_n \cdot x^n = c_0 + x \cdot (c_1 + x \cdot (c_2 + \cdots + x \cdot (c_n) \cdots))
\]

(b) \(\textit{findJump} :: \text{Integer} \rightarrow [[[\text{Integer}]]) \rightarrow (\text{Integer}, \text{Integer})
\)
takes an integer \(d\) and a list and returns the first pair of adjacent elements of the list such that the values of these two elements are farther than \(d\) apart, e.g.,
\(\textit{findJump} 3 [2,3,4,2,5,3,6,2,3,5,4,1,6] = (6,2)
\)
If the list contains no such values, an error is produced.

(c) \(\textit{suffixes} :: [a] \rightarrow [[[a]]]
\)
delivers for each argument list all its suffixes, e.g.:
\(\textit{suffixes} \{1,2,3,4\} = [[[1,2,3,4], [2,3,4], [3,4], [4]], []] \)
(The order is irrelevant.)

(d) \(\textit{diagonal} :: [[[a]]] \rightarrow [a]
\)
interprets its argument as a matrix (represented as in Exercise 2.1), which may be assumed to be square, and returns the main diagonal of that matrix, e.g.:
\(\textit{diagonal} [[[1,2,3], [4,5,6], [7,8,9]] = [1,5,9]
\)

(e) \(\textit{isSquare} :: [[[a]]] \rightarrow \text{Bool}
\)
determines whether its argument corresponds to a list-of-lists representation (as in Exercise 2.1) of a \textit{square} matrix.
Exercise 4.5 — Haskell Evaluation (30% of Midterm 1, 2003)

Assume the following Haskell definitions to be given:

- \text{foldr} :: (a \to b \to b) \to b \to [a] \to b
- \text{foldr} f e \text{[]} = e
- \text{foldr} f e (x:xs) = f x (\text{foldr} f e xs)
- \text{concat} = \text{foldr} (++) \text{[]}
- \text{(||)} :: \text{Bool} \to \text{Bool} \to \text{Bool} -- \text{Boolean disjunction: or}
- \text{True} || _ = \text{True}
- \text{False} || b = b
- \text{any} p = \text{foldr} ((||) . p) \text{False}
- \text{gen} f (x,s) = x : \text{gen} f (f x s)
- \text{foo} k n = (k + n, n + 2)

Simulate Haskell evaluation for the following expressions (write down the sequence of intermediate expressions):

(a) \text{foldr} (*) 1 \text{[6,7]}

(b) \text{any} (> 0) \text{(gen foo (0,1))}

Exercise 4.6 — Defining Haskell Functions (20% of Midterm 1, 2004)

Define the following Haskell functions (the solutions are independent of each other):

(a) \text{sum} :: [\text{Integer}] \to \text{Integer}

such that \text{sum} \text{xs} evaluates to the sum of all elements of the list \text{xs}.

(b) \text{all} :: (a \to \text{Bool}) \to [a] \to \text{Bool}

such that \text{all} \text{p} \text{xs} evaluates to \text{True} if \text{p} considered as a predicate holds for all elements of \text{x},
and to \text{False} if there is at least one element in \text{xs} for which \text{p} does not hold.

E.g., \text{all} (> 1) \text{[2..10]} = \text{True}

(c) \text{selMod} :: \text{Integer} \to [\text{Integer}] \to [\text{Integer}]

such that \text{selMod} \text{k} \text{xs} selects from the list \text{xs} all those elements that are equivalent to \text{k} modulo \text{k} + 1, e.g.,

\text{selMod} 2 \text{[2, 3, 8, 1, 2, 5]} = [2, 8, 2, 5]

(d) \text{sources} :: \text{Eq} \text{a} \Rightarrow [(a,a)] \to [a]

such that \text{sources} \text{ps} returns the \text{sources} of the graph \text{ps}.

Here, the list \text{ps} of pairs is considered as representing a simple graph by representing each edge from node \text{x} to node \text{y} by the pair (\text{x}, \text{y}).

The context “\text{Eq a ⇒}” just means that you may use the equality test for elements of type \text{a},
i.e., (==) :: a \to a \to \text{Bool}.

Example: \text{sources} \text{[(2,3), (3,4), (1,4), (1,5), (2,5)]} = [2,1]

(The order is irrelevant.)