ASSIGNMENT 5A – HOF ON LISTS

Advanced programming paradigms

In this assignment, you will apply different higher-order functions on lists to gain experience in those typical functional-oriented operations.

Question 1 – *Higher-order functions on lists*

Using only the `zip` and `map` higher-order functions, define the following functions that:

(a) Returns a list of the length of each string of a list
Example:
`lengthStrings(List("How","long","are","we?"))` returns `List(3,4,3,3)`

(b) Produces a list with n identical elements of arbitrary type (don't use the `fill` method!)
Examples:
`dup("foo", 5)` returns `List("foo", "foo", "foo", "foo", "foo")`
`dup(List(1,2,3), 2)` returns `List(List(1,2,3), List(1,2,3))`

(c) Multiplies element-wise two lists of values and create a new list
Example:
`dot(List(1,2,3), List(2,4,3))` returns `List(2,8,9)`

Question 2 – *Folding functions on lists*

Now using folding (right or left) define a function that:

(a) Determines if all logical values in a non-empty list are `true`.
Examples:
`areTrue(List(true, true, false))` returns `false`
`areTrue(List(true, true, true))` returns `true`

(b) Determine the total length of the strings in a list
Example:
`lString(List("Folding", "is", "fun"))` returns 12

(c) Returns the longest string of a list as well as its size
Example:
`longest(List("What", "is", "the", "longest?"))` returns `(8, longest?)`

(d) Decide if a value is an element of a list of an arbitrary type
Examples:
`isPresent(List(1,2,3,4), 5)` returns `false`
`isPresent(List(1,2,3,4), 3)` returns `true`

(e) Flatten a nested list structure of any type.
Example:
`flattenList(List(List(1,1), 2, List(3, List(5, 8))))` returns `List(1,1,2,3,5,8)`