Lab 08: Recursion

Spring 2018

1  Question 1

Create a function order? that consumes a listof Num, which has length at least two, and that does not contain any duplicates. It returns "ascending" if the numbers are ordered in ascending order, "descending" if it is in descending order, and "mixed" if neither. For example
(order? (list 1 3 4 7)) => "ascending"
(order? (list 1 2 3 6 5)) => "mixed"

2  Question 2

Create a function rid that consumes a list of at least length 2 and gets rid of the first and last items of a list and keeps the items in the middle. For example (rid (list 1 2 3 4)) => (list 2 3).

3  Question 3

Create a function more-even? that consumes a non-empty list of natural numbers and returns true if there are more even numbers than odd numbers in the list.

4  Question 4

Create a function (multi n L) that consumes a Nat and a listof Num and returns a list containing all the values of L, where the first n values have been multiplied by 10. For example:
(multi 3 (list 2 3 5 7 11)) => (list 20 30 50 7 11).

5  Question 5

Create a function (double-nums L) that consumes a listof Any and returns a copy of the list where all values in L that are numbers have been multiplied by 2. The last value in the list is not changed, even if it is a number. For example
(double-nums (list 1 "a" 2 "b" 3)) => (list 2 "a" 4 "b" 3)