BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
PILANI CAMPUS
FIRST SEMESTER 2020 – 2021
PRINCIPLES OF PROGRAMMING LANGUAGES (CS F301)
Tutorial (6th November 2020)

---

1) Write the tail recursive program f_reverse : : [a] -> [a] to reverse the list in Haskell.

Examples:

f_reverse [1,2,3] = [3,2,1].
f_reverse [] = []
f_reverse [True, False] = [False, True]

2) Define a function f_remove :: [Int] -> [Int] that removes adjacent duplicates. i.e., if the same element occurs n times contiguously, we retain only one copy.

Examples:

f_remove [1,1,1,2,2,3,3,3,3] = [1,2,3]
f_remove [1,2,1,2,3,1,1,2,2] = [1,2,1,2,3,1,2]

3) Define a function f_subSeq :: String -> String -> Bool which checks whether the first argument is a subsequence of the second. A subsequence is obtained by deleting some letters in a string and retaining the other characters in the same order as in the original string.

Examples

f_subSeq "ab" "abc" = True
f_subSeq "ab" "acb" = True
f_subSeq "ab" "bca" = False
f_subSeq "" "bea" = True
f_subSeq "ba" "ba" = True

4) A two-dimensional matrix can be represented as a list of rows, each row itself being a list of elements. So in general it is of type [[a]]. Not every list of lists is a matrix, though. For instance, [[1,2,3], [], [2,4]] is a list of three lists, each of a different size.

   a. Define a function f_isMatrix :: [[a]] -> Bool that checks if a list of lists is a valid matrix (nonzero number of rows, each of the same nonzero length).

b. A square matrix is one where the number of rows is equal to the number of columns. Define a function f_isSquareMatrix :: [[a]] -> Bool that checks if a list of lists is a square matrix.

Examples:

f_isMatrix [] = False
f_isMatrix [[],[],[]] = False
f_isMatrix [[2,3], [4,5], [6,7]] = True
f_isMatrix [[2,3,4,5,6,7]] = True

f_isSquareMatrix [] = False
f_isSquareMatrix [[]] = False
f_isSquareMatrix [[1]] = True
f_isSquareMatrix [[1,2,3],[4,5,6],[7,8,9]] = True
f_isSquareMatrix [[1,2,3,4],[5,6,7,8],[9,10,11,12]] = False

5)

a. Two matrices are addable if they have the same number of rows and same number of columns. Define a function f_addable :: [[a]] -> [[a]] -> Bool that checks if two matrices are addable.

b. Matrix m1 is multiplyable with matrix m2 if the number of columns in m1 is the same as the number of rows in m2. Define a function f_multiplyable :: [[a]] -> [[a]] -> Bool that checks if matrix m1 is multiplyable with m2.

Examples:

f_addable [[1,2],[3,4]] [[1,2],[3,4]] = True
f_addable [[1,2],[3,4]] [[5,6,7],[8,9,10]] = False
f_addable [[1,2],[3,4]] [[1,2],[3,4],[3,4]] = False

f_multiplyable [[1,2,3],[4,5,6]] [[1,2],[3,4]] = False
f_multiplyable [[1,2,3],[4,5,6],[1,2,3],[4,5,6]] [[1,2],[3,4],[5,6]] = True