

Programming with Lists Take Write an Oz function which takes two arguments, a list and a number and evaluates to the first elements of the list if n is a positive number less than the list length, to nil if n is either 0 or negative, and evaluates to the whole list if the list is shorter than n .

Drop Write an Oz function which takes two arguments, a list and a number and evaluates to the last elements of the list if n is a positive number less than the list length, to the whole list if n is either 0 or negative, and evaluates to nil if n is longer than the list.

Merge Write an Oz function which takes two sorted lists of integers as arguments, and evaluates to a merged list in sorted order. The two lists need not be of equal length.

Higher-Order Programming ZipWith Write an Oz function which takes 3 arguments - the first, a 2-argument function followed by two lists, and , and outputs the list whose i^{th} position is got by evaluating on the i^{th} elements in and

Map Using FoldR Rewrite using

FoldL Write an Oz function which folds a binary operation from the left. For example, {FoldL Sum [1 2 3] 0} should evaluate to {Sum {Sum {Sum 0 1} 2} 3}.