1. Define a matrix to be a list of lists of real numbers all of equal length. Write ML functions to implement addition and multiplication of matrices.

The following is based on Exercise 3.41 from *ML for the Working Programmer*:

2. Use the following function to code an alternative version of merge-sort that doesn’t use the `take` and `drop` functions. How does it compare in speed with the version given in the lecture? What accounts for the difference?

   ```ml
   fun alts ([], l1, l2) = (l1, l2)
   | alts ([x], l1, l2) = (x::l1, l2)
   | alts (x::y::t, l1, l2) = alts (t, x::l1, y::l2);
   ```

3. Write definitions in ML for the following functions:

   ```ml
   nodups : ''a list -> ''a list
   nodups l is the list containing all the elements of l, but without repetitions. So nodups [1,2,3,2] = [1,2,3].

   union : ''a list * ''a list -> ''a list
   which forms the union of two lists, i.e. the list containing all the elements of the two input lists, again without repetitions.
   ```

   Why were equality type variables used in the above specifications? Can you define similar functions for arbitrary types?