Matrices can be represented by a list of rows, where each row is represented by a list of the elements (going left to right). For instance, the matrix
\[
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{pmatrix}
\]
is represented by \([[1,2,3], [4,5,6]]\).

1. Define a function `diagonal` that takes a square matrix and returns its diagonal. You need not consider the case where the input list does not represent a square matrix.

   ```
   - diagonal;
   val it = fn : 'a list list -> 'a list
   - diagonal [[10]];
   val it = [10] : int list
   - diagonal [[1,2], [3,4]];
   val it = [1, 4] : int list
   - diagonal [[1,2,3], [4,5,6], [7,8,9]];
   val it = [1, 5, 9] : int list
   ```

   Any other function you define must be local to the above function.

2. Define a function `unitp` that will check whether the given list of lists represents an identity matrix.

   ```
   - unitp;
   val it = fn : int list list -> bool
   - unitp [];
   val it = false : bool
   - unitp [[5]];
   val it = false : bool
   - unitp [[1,0], [0,1]];
   val it = true : bool
   - unitp [[1,2], [0,1,1]];
   val it = false : bool
   - unitp [[1,2], [0,1], [0,0]];
   val it = false : bool
   ```

   Any other function you define must be local to the above function.