1. Suppose that we represent integer matrices (i.e., matrices whose entries are integers) 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 \([[1,2,3], [4,5,6]]\).

Define a function \texttt{unitp} that will check whether the given list of lists represents an \textit{identity matrix}.

For instance,

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

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

2. Define a function \texttt{foo} which takes a list \texttt{ls} and checks whether it is a concatenation of two identical lists, i.e., whether there exists \texttt{L} such that \texttt{ls} = \texttt{L@L}. (Any other function you define must be local to the above function.)

- \texttt{foo};
  val it = fn : ''a list -> bool
  - \texttt{foo nil};
  val it = true : bool
  - \texttt{foo [true, true]};
val it = true : bool
- foo [1,2,3];
val it = false : bool
- foo [1,2,1,2];
val it = true : bool