In this problem, we’ll implement a fragment of a program that checks solutions of Sudoku puzzles. A Sudoku board is a 9x9 matrix divided into nine 3x3 boxes. Initially, the board is partially filled with numbers from 1 to 9, with some squares left blank. For example:

\[
\begin{array}{|c|c|c|}
\hline
2 & 1 & 3 & 8 \\
\hline
7 & 6 & 1 & 3 \\
\hline
9 & 8 & 1 & 2 & 5 & 7 \\
\hline
3 & 1 & 8 \\
\hline
9 & 8 & 2 \\
\hline
5 & 6 & 9 & 7 & 8 & 4 \\
\hline
4 & 2 & 5 \\
\hline
\end{array}
\]

To solve the puzzle the player must fill in the remaining squares of the matrix so that:

- no row contains the same number twice
- no column contains the same number twice
- no 3x3 box contains the same number twice

We represent a board as a list of lists of characters: `List[List[Char]]`. There are nine `List[Char]` in the board, each of which represents a row. Each element is one of the Char `‘1’` through `‘9’`, plus `‘.’` to represent a blank square. The above board can be constructed as follows:

```scala
List("2....1.38",
    "........5",
    ".7...6...",
    "......13",
    ".981..257",
    "31....8..",
    "9..8....2.",
    ".5..69784",
    "4..25....").map(_.toList)
```

To get a list of all rows of the board, we can just write the identity function:

```scala
def rows(board: List[List[Char]]) = board
```
In the following questions, you may use any methods of the Scala library to assist you.

(a) [5 pts] Write a function `cols` that returns a list of the columns of the board. Each column is a `List[Char]`. Your function should have the signature:

```
def cols(board: List[List[Char]]): List[List[Char]]
```

*Hint:* There is a method of `List` that does exactly what’s required.

(b) [5 pts] The function `boxes` that returns a list of the 3x3 boxes of the board:

```
def boxes(board: List[List[Char]]): List[List[Char]] =
  board.map(_.grouped(3)).grouped(3).map(_.transpose).flatten.map(_.flatten)
```

Explain how this function works. Be precise.

(c) [5 pts] Write a function `nodups` that given a `List[Char]`, returns `true` if the list has no duplicates.

(d) [5 pts] Write a function `correct` that given a `List[List[Char]]`, returns `true` if the board is correct according the the three rules enumerated above.