1. **Helper function "buildDataEnvs"**
   This function takes a list of value columns and an environment, returns a list of environments corresponding to each row. First it creates a list of empty environments with the same length of the list of values(since that is how many rows the spreadsheet has) as the initial value of "foldl". Then, for each column in the argument list, it generates a list of pairs which has the form [(value of a row, environment of the same row), …] through the use of "zip". Last, it uses the function "map" to map each pair in the list to a new environment by inserting the column name and the value to the environment for each pair. After foldl iterates through all the columns, we have the desired environments.
   Helper function "buildValColByFormula"
   This function takes a formula from a formula column(which is a DeerLang expression) and a list of environments for rows, and returns a list of values. The helper function simply uses "map" to map each environment of a row to a value of the corresponding row, using "evalDeer" to evaluate the formula with that environment. After we evaluate the formula using a different environment for each row, we get a list of values, one corresponding to each row.
   Helper function "getNumRows"
   This helper function takes a value column, and returns the number of rows for that column. The function returns the length of the list of values of that value column since that is the number of rows.
2. In my opinion, pattern matching definitely provides a lot of convenience in this project. Specifically, it allows me easily decompose a data type in order to grab necessary information for further process. Also, it allows me to check input errors like incorrect numbers of arguments. Moreover, it is easier to debug given that Haskell is strongly-typed. A challenge about using Haskell vs an imperative language is that sometimes it is hard to find a straightforward way to update a variable(ie. Update the environment as a formula column is evaluated).
3. I work as a team for this project. However, I wrote all the code in P1.hs as well as the majority of testing since my partner is mostly occupied by other course work and responds fairly slow.