QUESTION 1

Write a Haskell implementation of the old Animals guessing game. Start by prompting "Think of an animal. Hit return when ready." Wait for the user to hit return, then ask: "Is it a penguin?" and wait for a Y or N answer. If the answer is yes, print out that you guessed it with 0 questions. If no, then ask them what their animal was, ask them to enter a yes or no question that would distinguish their animal from a penguin, and whether the answer is yes or no for their animal.

Then start over. This time start by asking them the question they just entered, and depending on their answer, and the answer they said to expect for their previous animal, ask them if their (new) animal is their previous animal, or ask if it is a penguin. If that is correct, print out that you guessed it with 1 question. If no, then ask them what their animal was, ask them to enter a yes or no question that would distinguish their animal from the one you just guessed, and whether the answer is yes or no for their animal.

The game proceeds like this indefinitely. You should build up a decision tree with questions at the nodes, and animals at the leaves. For each animal they think of, you traverse the tree from the root asking questions and following the branch selected by their answer until you reach a leaf, then guess the animal at the leaf, and get them to give you a question to extend the tree if you get it wrong.