Assignment 2

1. Recall the `Shape` type:

   ```
   data Shape = Rectangle Float Float
              | Ellipse Float Float
   ```

   This is not self-documenting, since it does not say which parameter is the width and which one is the height. Fortunately, there is syntactic support for it. Read section 3.15 and possibly 4.2.1 of the Haskell Report on labelled fields, and rewrite `Shape` with labelled fields.

2. Given this declaration of a binary tree type, where each internal node bears a piece of data:

   ```
   data Tree a = Nil
               | Branch a (Tree a) (Tree a)
   ```

   write a `mapTree` function

   ```
   mapTree :: (a->b) -> Tree a -> Tree b
   ```

   `mapTree f t` returns a new tree `t'` that has the same structure as `t`, but the data in every node of `t'` is obtained from applying `f` to the data at the corresponding node in `t`.

3. The emergency room in a hospital is not first-come first-served, for obvious reasons; instead, a nurse assesses the condition of every arriving patient and assigns him/her a priority number accordingly. Whenever the doctors are ready for the next patient, the patient with the smallest priority number gets treatment.

   The hospital adminstration wishes to computerize part of this process to aid the nurse. Because of the safety-critical nature of this task, you are asked to implement it in Haskell. (Yeah right.) The program lets the nurse enter a patient's name and priority, get the next patient for treatment, and close down the program.

   The input consists of a bunch of lines, each being one of these:

   - `A Alice 5`

     This means a patient Alice has arrived and is assigned priority 5. Your program should store the data.

   - `N`

     This means the doctors are ready and the nurse asks your program for the next patient. Your program should output on a line the name of the patient to be treated (and remove his/her entry from your store.) If there are several patients with the same smallest priority number, the one who arrived earliest gets treatment.

- `Q`
  This means the nurse closes down your program. Your program should exit. There are no more input lines after this.

The output is as specified above.

Bonus: you may also like to process this:

- `L`
  The nurse asks your program to list all patients in its store, from the smallest priority to the largest priority. Like in the above, if several patients have the same priority number, order them from the earliest to the latest.

Sample Input:

```
A Alice 5
A Bob 4
N
A Charles 5
A Dennis 3
N
N
N
Q
```

Sample Output:

```
Bob
Dennis
Alice
Charles
```

To save writing, source code for a leftist heap module, a red-black tree module, and a scheduled queue module will be made available on the web page, and you can use them. Choose the right data structure for simplest coding and best performance!