4

The National Society of Zoo Electricians needs to construct a database of the electrical properties of zoo animals. They have already got a standard for the object that will represent the individual animals:

```
struct zanimal
{ string name;        // e.g. “Squeaky”, “Roary”, “Oinky”, “Fluffy”, …
  string species;     // e.g. “Bat”, “Tiger”, “Seal”, “Penguin”, …
  float resistance;   // in Ohms
  float capacitance; // in Farads
  float mrv;          // Maximum Rated Voltage, in Volts

  /* a constructor has also been defined */ };
```

It is your job to build the database as a vector (or flexible, growable array).

There are two vital rules:

1. The vector must be given a fully object oriented definition, with a constructor and methods and appropriate use of public and protected.
2. Speed of execution is important. More important than minimising memory use.

These are the requirements:

1. Must be able to create an empty collection and add an unlimited number of zanimal objects to it with code like this:

```
zlist L;
L.add(new zanimal("Mike", "Brown Bear", 455.0, 0.0035, 450));
L.add(new zanimal("Eric", "Eel", 270.0, 0.00065, 1100));
L.add(new zanimal("Martha", "Moth", 945.0, 0.021, 240));
```

2. Must be able to find all the information (i.e. the entire zanimal object) for the animal that has the lowest resistance of all of a given species in the collection, like this:

```
most_conductive_moth = L.min_resist("Moth");
```

3. Must be able to find the average resistance of all the animals in the collection, like this:

```
double avg_res = L.average_resistance();
```

a.

Taking into account the two rules and three requirements, write all the class and method definitions in C++.

b.

Why do you think the NSZE used floats in their definition of a zanimal?

5

A binary tree for storing strings in alphabetical order is defined by a struct that begins like this:

```
struct treenode
{ string info;
  treenode * left, * right;
```

a.

The following strings are to be inserted into an initially empty tree, in exactly the order shown:

i. pig
ii. yak
iii. rat
iv. cow
v. toad
vi. hog
vii. man
viii. dog
ix. ant
x. eel
xi. flea
xii. western-reticulated-ground-squirrel-with-one-leg-missing

Draw a diagram showing exactly the shape of the tree as it is after each insertion. There are twelve insertions, so I should see twelve diagrams.

Make sure to show all the nodes and pointers clearly. When a node has only one pointer, make sure it is very clear whether it is the left or right pointer.

b.

Define the C++ method or function for inserting new strings into trees. Be sure that your function would produce the results that you drew for part a.

c.

Write a method or function that would find (and return as its result) the longest string in a tree.

In case you have forgotten, if you have a string in a variable called `s`, the length of that string is delivered by `s.length()`.