8.

The following struct is used to represent magnetic animals

```
struct manimal
{ string name, species;
  int weight;

  manimal(string n, string s, int w)
  { name=n; species=s; weight=w; } };
```

Magnetic animals are to be stored in a Binary Tree, ordered by their names.

A.

Define the necessary struct(s) together with an `add` method, so that this sample code would correctly build a tree containing six animals:

```
ManiTree zoo;
zoo.add(new manimal("Lenny", "lion", 320));
zoo.add(new manimal("Sammy", "seal", 135));
zoo.add(new manimal("Quacky, "puppy-eating duck", 5));
zoo.add(new manimal("Timmy", "tiger", 251));
zoo.add(new manimal("Rosie", "pony", 260));
zoo.add(new manimal("Geoffrey", "georaffe", 1326));
```

```
manimals are already defined, no freedom there, have to take them as they are.
Example shows zoo.add takes a pointer to a manimal, so even if we had forgotten
that we should almost always use pointers in trees, again that gives us no
choice. An animal is not a node in a tree, and a node in a tree is not a whole
tree. Therefore need three distinct object types, one already given to us.

struct maninode
{ manimal * ani;
  maninode * left, * right;

  maninode(manimal * a)
  { ani=a; left=NULL; right=NULL; } };

struct manitree
{ maninode * root;

  manitree()
  { root=NULL; }

  void add(maninode * & current, manimal * addition)
  { if (current==NULL)
      current = new maninode(addition);
    else if (addition->name < current->ani->name)
      add(current->left, addition);
    else
      add(current->right, addition); }

  void add(manimal * addition)
  { add(root, addition);

  /* answer to part C will go here */ };
```

B.

Draw the tree, showing nodes connected by pointers, as it would be directly after the sample code is executed.

```
manitree is in pink
maninodes are in yellow
manimals are in grey
circles are for NULLs
```

```
The important thing is the structure of the yellows. Don't get distracted by
the greys, they are not part of the tree structure, just the data that it
carries.
```

root:

lenny, ...

sammy, ...

geoff...

quacky.

timmy

rosie...
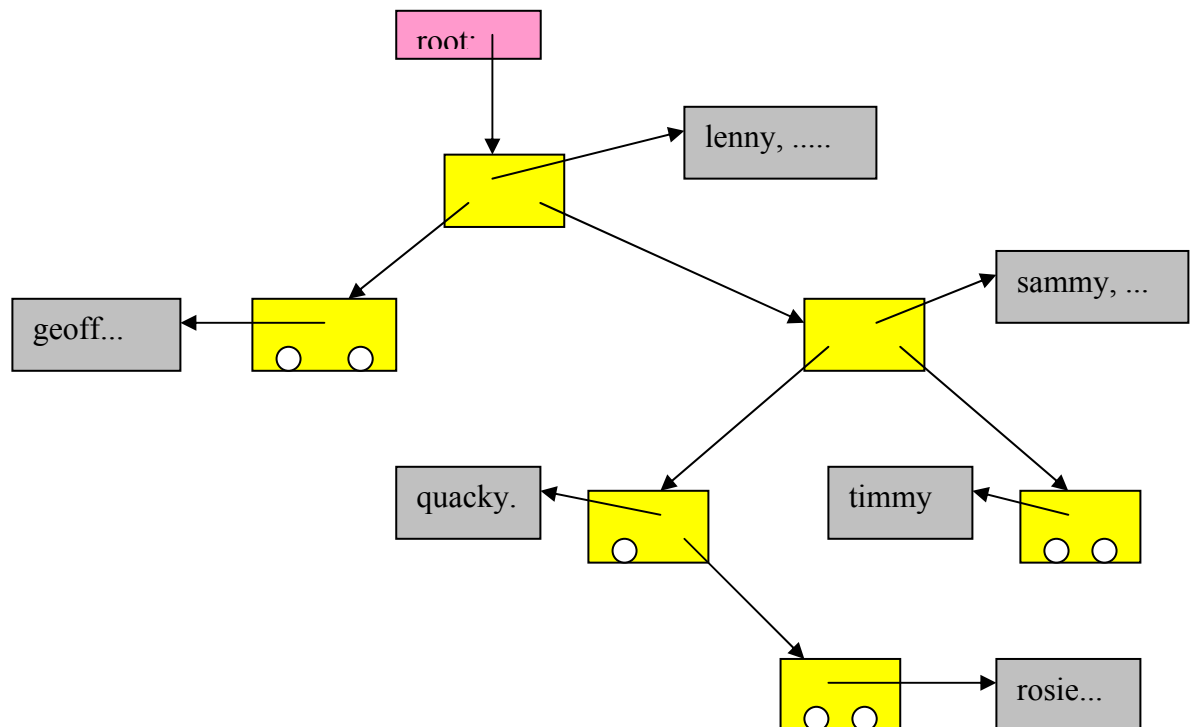
C.

Add a method to the tree-of-animals class. It should take an animal's name as its parameter, and return as its result that animal's weight, or -999 if there is no match. Your method should not print the result itself, just return it. So

```
    int w = zoo.weightof("Timmy");
    cout << "Timmy weighs " << w << " pounds";
```

should print

```
    Timmy weighs 251 pounds
```

```
/* remember this is a method of manitree */

The simplest way is probably to make an ordinary find method that
finds a manimal in the tree, then use that to do the real work in
weightof

To get from a pointer to a node to the name of an animal, you have to
follow two pointers. One to get from the pointer to the node, then
another to get from there to the animal.

  manimal * find(string wanted)
  { maninode * ptr = root;
    while (true)
    { if (ptr == NULL)
        return NULL;
      else if (ptr->ani->name == wanted)
        return ptr->ani;
      else if (ptr->ani->name < wanted)
        ptr = ptr->left;
      else
        ptr = ptr->right; } }

Now weightof is a piece of cake

  int weightof(string name)
  { manimal * it = find(name);
    if (it == NULL)
      return -999;
    else
      return it->wieght; }
```

9.

Show your reasoning, use human-oriented units such as hours, months, or
years rather than enormous numbers of seconds. State any assumptions that
you make in answering these questions.

A.

Using the O(...) notation, and N to represent the number of data items
involved, what is the speed of:

i.   Insertion sort   `O(N²)`

ii.   Merge sort   `O(N log N)`

iii. Bubble sort   `O(N²)`

iv.   Binary Chop Search   `O(log N)`

v.   Searching a Linked List `O(N)`

vi.   Searching a well balanced Binary Tree   `O(log N)`

B.

If an implementation of Insertion Sort on a particular computer is found to take 1,000 seconds to sort 1,000,000 items:

```
insertion sort is O(N²), N is 10⁶, so it performed around 10¹²
ops in 1,000 seconds, so speed is 10⁹ ops per sec.
```

i.   How long would you expect it to take to sort 3,000,000 items?

```
(3×10⁶)² is 9×10¹² or about 10¹³, speed is 10⁹ ops per sec,
therefore it takes about 10⁴ seconds. 10,000 seconds is a bit less
than 3 hours.
```

ii.   How long would you expect it to take to sort 1,000 items?

```
(1000)² is 10⁶ ops, 10⁶÷10⁹ is 10⁻³ so 0.001 sec or 1mS.
```

iii.   How long would you expect it to take to sort 1,000,000 items?

```
(1000000)² is 10¹² ops, 10¹²÷10⁹ is 10³ so 1000 secs or ¼ hour.
```

iv.   How long would you expect it to take to sort 1,000,000,000 items?

```
(10⁹)² is 10¹⁸ ops, 10¹⁸÷10⁹ is 10⁹ so 10⁹ secs 1000 million seconds.
One year is 30 million seconds, 30×30 is about 1000, so 30 years.
```

C.

If it takes one tenth of a second to search a linked list of 1,000,000 items, and those items were then put into a binary tree, how long would you expect a search of that tree to take?

```
linked list search is linear, so 1,000,000 ops in 0.1 second,
or 10⁷ ops per second for the computer speed. Searching for
something in a tree is logarithmic, log₂(1,000,000) is 20 ops.
20 ops at 10⁷ per sec is 2 micro-seconds.
```

D.

If it takes Bubble Sort an hour to sort an array of 1,000,000 items,

```
bubble sort is O(N²), N is 10⁶, so it performed around 10¹² ops
in 3600 secs, so speed is around 3×10⁸ ops per sec.
```

i.   How long would it take for a Binary Chop Search to find an item in the resulting sorted array?

```
binary chop sch is logarithmic, log of 1,000,000 is still 20,
so 20 ops at 3×10⁸ per second, around 7×10⁻⁸ secs or 70 nano-
seconds.
```

ii.   How long would it have taken to sort the array if Merge Sort had been used instead of Bubble Sort?

```
merge sort is N log N, so 20,000,000 ops. 2×10⁷ ops at 3×10⁸
per second is about 7×10⁻² secs or 70 milli-seconds.
```

E.

You may not have heard of "Boolean Satisfiability", but its speed is $O(2^n)$. That is, Exponential.

   If it takes Boolean Satisfiability 1mS for 10 items, how long would it take for 100 items?

```
2^10 is about 1,000 ops performed in 1/1000 second, so a speed
of 10^6 ops per second.

2^100 is 2^10×2^10×2^10×2^10×2^10×2^10×2^10×2^10×2^10×2^10 so as 2^10 is 1,000 it
must be that 2^100 is 1,000,000,000,000,000,000,000,000,000,000
or 10^30. 10^30 ops at 10^6 per second gives 10^24 seconds. One year
is 3×10^7 seconds so it comes to 3×10^16 or thirty thousand
million million years.
```

10.

What is the difference between a crocodile and an alligator?
   - a.   one is bigger than the other
   - b. ☑   they are both the same size
   - c.   crocodiles do not like potatoes
   - d.   because one back leg was both the same
   - e.   none of the above

Who discovered Christopher Columbus?
   - a. ☑   America
   - b.   a Western route to India
   - c.   Viking settlers
   - d.   Magellan
   - e.   Vasco da Gama

When was 1815?
   - a.   the battle of Waterloo
   - b.   Napoleon
   - c.   1776
   - d.   the battle of New Orleans
   - e. ☑   Belgium

`nobody got Belgium right.`

Where do hippopotamusses generally build their nests?
   - a. ☑   in mighty oak trees
   - b.   like an allegory on the banks of the Nile
   - c.   where angels fear to tread
   - d.   in the ashes of broken dreams
   - e.   illustrate your answer with an illustration