EEN218
Happy Lucky Second Test
17th April 2017

No books, notes, calculators, or phones.
No puppies, not even a little one.

Who are you?

What is your student number?

Did you cheat on this test?

Sign that statement.

Don't make any marks in my boxes.
These are my boxes.

Question	5	6	7	8
Out of	33	33	33	1
Grade				

5. Fast Sorting

Of course you can write extra functions that make the job easier.

a. Write a function that splits a linked list into two halves that are as close to equal in length as possible.

b. Show how this function can be used to make a sorting algorithm.
(don't just write the rest of the code; explain).

6. Vectors.

Implement a vector of `person` objects. For each person record their first name, last name, and I.Q.

Your implementation must include at least the following:

a. A constructor
b. A `push_back` method
c. A `pop_back` method
d. A `grow_to` method
e. A `shrink_to` method

`grow_to(N)` ensures that the capacity of the vector is at least N, resizing if necessary. `shrink_to(N)` ensures that the capacity of the vector is no more than N, resizing if necessary.

7. Basic Binary Trees.

Just in case you ever own a chicken farm, you're going to make a binary tree to store all their chickenly records. For each chicken, the following pieces of information will be stored: first name, last name, birthday, weight (in ounces). Yes, modern chickens have two names, just like people.

a. Define a struct suitable for representing a chicken.

b. Define a struct and its constructor suitable for representing a node in a binary tree of chickens.

c. Add a function that will add a new chicken to a tree.
The tree should be ordered alphabetically on the chickens' last names. If two chickens have the same last name, break the tie by using their first names.

d. Add a function that will search a tree to find a chicken with given first and last names (parameters).
Be sure that your function behaves in a reasonable way if no such chicken can be found.

8. The End.

Draw a picture of Question 7.