1. (3 pts) Assuming that we are implementing a member function which will return the tail of a linked list, what is the appropriate header for the function as defined outside of the class (not inline), choose the best answer.
   a. LList* LList::getTail(LListNode* head)
   b. LList LList::getTail(LListNode* head)
   [✓ c. LListNode* LList::getTail(LListNode* head)]
   d. LListNode* LListNode::getTail(LListNode* head)

2. (3 pts) Given two classes, Base and Derived where Derived derives from Base (obviously) both of which have a function, func, appropriately defined, which of the following can be used inside Derived's func to call the base class version of the function?

   a. func();

   b. this->func();

   [✓ c. Base::func();]

   d. super.func();

3. (3 pts) Given a base class pointer (Base * bptr) to a derived class object is the following allowed or not allowed?
   Base b = *bptr;
   [✓ a. Always allowed]
   b. Never Allowed
   c. Allowed if there exists a function to do so in the base class
   d. Allowed if there exists a function to do so in the derived class

4. (3 pts) Convert the math expression (2+3)+4*(6-5) to post fix form.

   23+465-*+

5. (3 pts) Evaluate the post fix expression "`3 2 * 3 - 2 /`" to a value.

   1.5 (if / means division)
   1 (if / mean integer division)

6. (15 pts) Given a binary search tree of ints, in which each node contains a size parameter (also an int), explain, in English, not code, how you would find the median element of the tree in theta(log N) time:

this is an ambiguous question. It can either be interpreted as "the size parameter stores the size of the tree whose node is the current node", or "the size parameter stores the *count* of the element in the current node - i.e. how many times the element has been repeated"

In scenario #2, it's not possible to get the median element in logrithemic time because you don't know how many elements there are in total.

In scenario #1,
1. let a point to a node (name it "curr") point to the root.
2. Calculate the median index:
   a. odd
   b. even

compare the size of right and left subtree. Go to the bigger subtree.

7. (15 pts) Given a pointer to the first node of a linked list, you are asked to reverse the list. Explain, in English, not code, how you would complete such a task if you had a limited amount of memory (not enough to store the whole list twice!)

   do an in-place reverse. Have two extra pointers prev and next, which stores the previous node and next node respectively. Initially, set prev to nullptr and curr to the head. Set next to curr->next, in order to keep track of the rest of the list. while..

8. (20 pts) Someone has given us a file (names.txt; it is guaranteed to exist) with a list of names, one per line. Each line in the file may contain more than one word. Some examples of names are as below:

```
John F. Jones
Jim Smith
Madonna
Paul Bunyon
Madonna
```

   Unfortunately, there exist duplicates in the file. Please write a program to remove the duplicate names and produce a file (cleanNames.txt) with only unique names.

9. (20 pts) Given a pointer to the root of a binary search tree (has left, right, and parent pointers as well as a data section) write a function (or functions) which will return an STL list (you should not define this class, it's already included) with all of the values from the tree in sorted order. Your code should run in theta(N) time.

10. (15 pts) Our company sells various types of furniture. We're going to design a program to handle our furniture inventory and you are being asked to design two classes. The base class will be "Furniture" and from it, you will derive a class "Chair." All Furniture has a function called "display" but what this function does is not known to the Furniture class. Each Furniture object will have to implement this function before it can be instantiated. Design the classes and provide a main function to put three chairs into a vector of Furniture object pointers.