Quiz 10 Answers

1. What is recursion?

   A general programming technique used to solve problems with a “divide and conquer” strategy. It is normally implemented by having a function call itself within the program text.

2. Write a code fragment (not a complete method) that will insert `newNode` at the front of a non-empty list.

   ```
   newNode->next = head;
   head = newNode;
   ```

3. Write a code fragment that will insert `newNode` at the rear of a non-empty list.

   ```
   tail->next = newNode;
   tail = newNode;
   ```

4. Write a code fragment that will insert `newNode` into the list after the node pointed to by `current`.

   ```
   newNode->next = current->next;
   current->next = newNode;
   ```

5. Write a recursive method that will return the size of the list. The method should take a pointer to a `Node<T>` and return an `int`. When the method is initially called, it will be passed the list’s `head` pointer.

   ```
   template <class T>
   int List<T>::size(Node<T>* p) const
      {
      if (p == nullptr)
         return 0;
      else
         return 1 + size(p->next);
      }
   ```

6. Write a non-recursive method that will return the size of the list. The method should take no arguments and return an `int`.

   ```
   template <class T>
   int List<T>::size() const
      {
      Node<T>* p;
      int count = 0;

      for (p = head; p != nullptr; p = p->next;
         count++;

      return count;
      }
   ```

7. Declare a pointer to a function that returns a `bool` and takes two `double` arguments.

```
bool (*ptr)(double, double)
```

8. Rewrite the contents of the following array once the array has been partitioned by the quicksort partition code used on Assignment 8.

26	10	67	30	18	54	41	28

Once the array has been partitioned around the pivot value 30, it will look like this:

28	10	26	18	30	54	41	67