Noah Buchen
Midterm CS362 Winter 2019
Question 2:

```
srand(time(NULL));  // Seed ONCE

void testRemoveAll() {

	//TEST #1
	//test on empty container
	container *emptyC = newContainer();
	//removeAll
	removeAll(1, emptyC);
	//asserts
	assert(get(1, emptyC) == 0);
	assert(size(emptyC) == 0);

	//TEST #2
	//test on container that is full of a single number
	container *repeatingC = newContainer();
	//fill with one random number
	int oneRand = random();
	//because no usage for the container was specified I went with a large boundary
	for (int i = 0; i < INT32_MAX; ++i) {
		add(oneRand, repeatingC);
	}
	//removeAll
	removeAll(oneRand, repeatingC);
	//asserts
	assert(get(oneRand, repeatingC) == 0);
	assert(size(repeatingC) == 0)

	//TEST #3
	//test on a container containing random numbers
	container *randomC = newContainer();
	//fill with random numbers
	for (int i = 0; i < INT32_MAX; ++i) {
		int rand = random();
		add(rand, randomC);
	}
	//create more random numbers until match is found with get
	int toRemove = random();
	while(get(toRemove, randomC) == 0){
		int toRemove = random();
	}
	//this insures that removeAll can best find instances of the number within the container not just
the edges
	//removeAll
	removeAll(toRemove, randomC);
```

```
        //asserts
        assert(get(oneRand, repeatingC) == 0);

}
```