Summer 2012 - Final Part 2 – Answer Key

1. Searching and sorting

(a) Indexes examined: 6, 10, 8, 9
Value returned: -10

(b) choice #2: `{-5, 17, 3, 23, 46, 8, 14, 12}`

(c) {33, 14, 3, 95, 47, 9, -42, 13}

split {33, 14, 3, 95} {47, 9, -42, 13}

split {33, 14} {3, 95} {47, 9} {-42, 13}

split {33} {14} {3} {95} {47} { 9} {-42} { 13}

merge {14, 33} {3, 95} {9, 47} {-42, 13}

merge {3, 14, 33, 95} {-42, 9, 13, 47}

merge {-42, 3, 9, 13, 14, 33, 47, 95}

2. Collections Mystery

a) {brick, plaster}

b) {blue, green, yellow}

c) {fruit}

d) {IPL, animal, cat, dog}
{animal, cat, dog, IPL} accepted too

3. Binary Tree Programming

```
public void evenLevels() {
	overallRoot = evenLevels(overallRoot, 1);
}

private IntTreeNode evenLevels(IntTreeNode node, int height) {
	if (node!= null) {

		node.left = evenLevels(node.left, height + 1);
		node.right = evenLevels(node.right, height + 1);

		if(node.right == null && node.left == null && height % 2 == 1){
			node = null;
		}
	}
	return node;
}
```

4. Linked List Programming

```
public int removeRange(int min, int max) {
	if (max < min) {
		throw new IllegalArgumentException();
	}

	ListNode current = front;
	int count = 0;
	while (front != null && front.data >= min && front.data <= max) {
		front = front.next;
		count++;
	}
	while (current != null && current.next != null) {
		if (current.next.data >= min && current.next.data <= max) {
			current.next = current.next.next;
			count++;
		} else {
			current = current.next;
		}
	}
	return count;
}
```