# ICS 211 Spring 2016 Exam 2, March 30th, 2016

Clearly write your name on the **front** of this exam.

This exam is closed-book. No calculators are allowed. There are a total of 100 points.

Be sure to answer all parts of each question.

**Question 1** (10 points): Write the **Stack<E>** interface.

**Question 2** (10 points): Implement the following method.

**public class LinkedStack<E> implements StackInterface<E> {**

**private LinkedNode<E> top;**

**public void push(E item) {**

**Question 3** (20 points): Implement the following methods.

**public class ArrayStack<E> implements StackInterface<E> {**

**private int top;**

**private E[] array;**

**public E pop() throws EmptyStackException {**

**public booelan empty() {**

**Question 4** (10 points): Write the **Queue<E>** interface.

**Question 5** (10 points): Implement the following method:

**public class LinkedQueue<E> implements Queue211<E> {**

**private LinkedNode<E> front;**

**private LinkedNode<E> rear;**

**public boolean offer(E e) {**

**Question 6** (10 points): Implement the following method:

**public class CircularArrayQueue<E> implements Queue211<E> {**

**private E[] data;**

**private int front;**

**private int rear;**

**public E remove() {**

**Question 7** (10 points): Write a **recursive** method **odd** that takes as a parameter a **LinkedNode<E>** and returns a **boolean**. If the parameter is null or refers to a linked list of even length, the method should return **false**. Otherwise the method should return **true**.

**Question 4** (10 points): Implement this method to perform a binary search. If the value is found, the method returns the index of the value, otherwise the method returns -1. The array **data** is sorted in ascending order.

**static int binarySearch(E[] data, E value, Compartor<E> c) {**

**Question 8** (5 points): What is the worst-case runtime of binary search on a sorted array of *n* items.

**Question 10** (5 points): A double-linked list requires the same amount of storage as that of a single-linked list.

True or False (circle one)