



King Saud University

College of Computer and Information Sciences

Department of Computer Science

**Data Structures CSC 212**

**Midterm Exam Solution- Spring 2018**

Date: 22/03/2018

Duration: 90 minutes

Question 1 ..... 30 points

1.  $2\ 9\ 3\ 1\ +\ *\ 5\ 4\ 3\ \%\ 1\ -\ -\ +\ >\ 35\ 14\ 8\ +\ =\ ||$ . [10]

+	*	%	-	-
4 9 2	36 2	1 5 36 2	0 5 36 2	5 36 2
+	>	+	=	
41 2	F	22 35 F	F F	F

2.  $4\ +\ (9\ -\ (3\ *\ 2))\ \% \ 3\ +\ 5\ *\ (2\ +\ (6\ /\ 3))\ -1$ . [11]

+	-	*	%	+
4 +	9 ( 4 +	3 - 9 ( 4 +	3 % 4 +	4 +
*	+	/	-	\$
5 * 4 +	2 ( 5 * 4 +	6 + 2 ( 5 * 4 +	24 -	23 \$

3. Preorder: 2. Inorder: 8. Postorder: 11. [9]

Question 2 ..... 35 points

1. [17]

```
public static <T> void swapAdj(Queue<T> q) {
    for (int i = 0; i < q.length() - 1; i += 2) {
        T e1 = q.serve();
        T e2 = q.serve();
        q.enqueue(e2);
        q.enqueue(e1);
    }
    if (q.length() % 2 == 1) {
        q.enqueue(q.serve());
    }
}
```

or

```
public static <T> void swapAdj(Queue<T> q) {
    for (int i = 0; i < q.length() / 2; i++) {
        T e1 = q.serve();
        T e2 = q.serve();
        q.enqueue(e2);
        q.enqueue(e1);
    }
    if (q.length() % 2 == 1) {
        q.enqueue(q.serve());
    }
}
```

2. [18]

```
public static Stack<Character> replace(Stack<Character> s, char a, char b) {
    Stack<Character> t1 = new LinkedStack<Character>();
    Stack<Character> t2 = new LinkedStack<Character>();
    while (!s.empty()) {
        t1.push(s.pop());
    }
    while (!t1.empty()) {
        char w = t1.pop();
        s.push(w);
        if (w == a) {
            t2.push(b);
        } else {
            t2.push(w);
        }
    }
    return t2;
}
```

Question 3 ..... 35 points

1. [20]

```
public void insertBefore(T e, int i) {
    Node<T> tmp = new Node<T>(val);
    if (i == 0) {
        tmp.next = head;
        head = tmp;
    } else {
        Node<T> p = head;
        for (int j = 0; j < i - 1; j++) {
            p = p.next;
        }
    }
}
```

```
    tmp.next = p.next;
    p.next = tmp;
}
current = head;
}
```

2. [15]

```
public boolean notRightChild(T e) {
    return notRightChild(root, e);
}
private boolean notRightChild(Node<T> t, T e) {
    if (t == null) {
        return true;
    }
    if (t.right != null && t.right.data.equals(e)) {
        return false;
    }
    return notRightChild(t.left, e) && notRightChild(t.right, e);
}
```