King Saud University	College of Computer and Information Sciences Department of Computer Science
Data Structures CSC 212	**Quiz 2**
Date: 00/00/0000	Duration: 20 minutes

Student ID:	Name:
Section:	Instructor:

1	2	Total/30

Question 1 . . . . 30 points

Write the method **public static** `<T>` **void** `cls(List<T>l,` **int** `k)` that performs `k` circular left shifts on the list.

Example 1. *If $l : A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$, then after calling* `l.cls(1)`*, $l : B \rightarrow C \rightarrow D \rightarrow E \rightarrow A$. If instead we call* `l.cls(2)`*, $l : C \rightarrow D \rightarrow E \rightarrow A \rightarrow B$.*

1. Line 1:
   - Ⓐ **for** `(`**int** `i = 0; i < k; i++){`
   - Ⓑ **while** `(!l.last()){`
   - Ⓒ **while** `(!l.empty()){`
   - Ⓓ **while** `(!l.full()){`
   - Ⓔ None
2. Line 2:
   - Ⓐ `l.findFirst();`
   - Ⓑ `l.findNext();`
   - Ⓒ `l.remove();`
   - Ⓓ `l.insert(l.retrieve());`
   - Ⓔ None
3. Line 3:
   - Ⓐ `l.findFirst();`
   - Ⓑ `T e = l.serve();`
   - Ⓒ `l.findNext();`
   - Ⓓ `T e = l.retrieve();`
   - Ⓔ None
4. Line 4:
   - Ⓐ `l.findNext();`
   - Ⓑ `l.update(e);`
   - Ⓒ `l.remove();`
   - Ⓓ `l.insert(e);`
   - Ⓔ None
5. Line 5:
   - Ⓐ **if** `(l.empty())`
   - Ⓑ **if** `(!l.full())`
   - Ⓒ **if** `(!l.last())`
   - Ⓓ **if** `(!l.empty())`
   - Ⓔ None
6. Line 6:
   - Ⓐ **while** `(!l.full())`
   - Ⓑ **while** `(!l.last())`
   - Ⓒ **if** `(!l.last())`
   - Ⓓ **while** `(l.last())`
   - Ⓔ None
7. Line 7:
   - Ⓐ `l.findFirst();`
   - Ⓑ `l.insert(e);`
   - Ⓒ `l.findNext();`
   - Ⓓ `l.remove();`
   - Ⓔ None
8. Line 8:

Ⓐ `l.update(e); }}`

Ⓑ `l.insert(l.retrieve()); }}`

Ⓒ `l.remove(); }}`

Ⓓ `l.update(l.retrieve()); }}`

Ⓔ None

Question 2 . . . . 10 points

Choose the correct answer (the element in **bold** is the current element):

```
private static <T> void f(DoubleLinkedList<T> l1, DoubleLinkedList<T> l2) {
  if (l1.last() || l2.first())
    return;
  if (l1.retrieve().equals(l2.retrieve())) {
    l1.remove();
    l2.remove();
  } else
    l1.findNext();
  l2.findPrevious();
  f(l1, l2);
}
```

1. If l1: **B** A A D B C, and l2: B B D D A **C**, then after calling `f`:

   Ⓐ l1: B A D **C**

   Ⓑ l1: B A **C**

   Ⓒ There is infinite recursion

   Ⓓ The code may throw an exception

   Ⓔ None

2. If l1: **A** A B B, and l2: B B A **A**, then after calling `f`:

   Ⓐ l1 becomes empty

   Ⓑ l1: **B**

   Ⓒ There is infinite recursion

   Ⓓ The code may throw an exception

   Ⓔ None