

5. Line 6:

King Saud University

College of Computer and Information Sciences

(A) for(int i = 0; i < cpt; i++)

Department of Computer Science

Data Structures CSC 212

Practice - Fall 2018

Write the method public static <T> void removeLast(List<T> 1, int k), user of the ADT List, which removes the last k elements of 1. Assume that 1 is **not empty** and that k >= 0 and valid (less or equal the list's length). public static <T> void removeLast(List<T> 1, int k) { 1. Line 2: (A) 1.findPrevious(); } (B) 1.remove(); } (A) 1.remove(); (C) 1.findLast(); } (B) int cpt = 0; (D) 1.findFirst(); } (C) 1.findNext(); (E) None (D) int cpt = 1; (E) None 6. Line 7: 2. Line 3: (A) 1.remove(); (A) 1.findLast(); (B) 1.findLast(); (C) 1.findFirst(); (B) 1.remove(); (D) 1.findNext(); (C) 1.findFirst(); (E) None (D) 1.findNext(); (E) None 7. Line 8: 3. Line 4: (A) for (int i = 0; i < k; i++) (B) for (int i = 0; i < cpt - k; i++) (A) while (!1.first()){ (C) for(int i = 0; i < cpt; i++) (B) while (1.last()){ (D) for(int i = 0; i <= cpt - k; i++) (C) while (!1.last()){ (E) None (D) for (int i = 0; i < k; i++){ (E) None 8. Line 9: 4. Line 5: (A) 1.findFirst(); (B) 1.findNext(); (A) 1.insert(1.retrieve()); (C) 1.update(null); (B) cpt++; (D) 1.remove(); (C) 1.insert(k); (E) None (D) 1.remove(); (E) None 9. Line 10:

CSC 212 Page 2 of 2

- (A) 1.update(null);
- (B) 1.remove();
- (C) 1.remove(); 1.findNext();
- (D) l.insert(l.retrieve());
- (E) None

Question 2 points

Write the method removeOddElems, member of the class LinkedList, that removes all the elements having an odd position (the position of the first element is 0). Assume that the list is **not empty**. The method signature is: **public void** removeOddElems().

Example 1. If $l: A \to B \to C \to D \to E$, then 1.oddElems() returns: $A \to C \to E$.

```
public void removeOddElems() {
...
...
```

- 1. Line 2:
 - (A) Node<T> p = head;
 - (B) Node<T> p = head.next;
 - (C) Node<T> p = current;
 - (D) Node<T> p = head.next.next;
 - (E) None
- 2. Line 3:
 - (A) while (p != null && p.next != null){
 - (B) while (p != null){
 - (C) while (p.next.next != null){
 - (D) while (current.next != null){
 - (E) None

- 3. Line 4:
 - (A) p.next = p;
 - (B) p = p.next;
 - (C) p.next = p.next.next;
 - (D) current.next = p;
 - (E) None
- 4. Line 5:
 - $(A) p = p.next;}$
 - (B) p.next.next = current.next;}
 - (C) p = p.next.next;
 - (D) p.next.next = p.next;}
 - (E) None