King Saud University
College of Computer and Information Sciences
Computer Science Department

CSC 212 **First Semester 1439-1440**

Tutorial # 7

Problem 1

1. Write the recursive static method ***copyStack*** that takes two Stacks *srcStack* and *destStack* and copies all the elements of *srcStack* into *destStack* in the same order while preserving *srcStack*. You can assume *destStack* can hold all *srcStack* elements. You are not allowed to use any auxiliary data structures.
   Method: *public static<T> void copyStack(Stack<T> srcStack, Stack<T> destStack)*

```
public static <T> void copyStack(Stack<T> srcStack, Stack<T>
destStack) {
     if (srcStack.empty())
          return;
     T elem = srcStack.pop();
     copyStack(srcStack, destStack);
     srcStack.push(elem);
     destStack.push(elem);
}
```

2. Write the recursive static method ***searchStack*** that searches for an element *elem* in a Stack *stack* and returns true if it's found or false otherwise. *stack* should not change at the end of the method. You are not allowed to use any auxiliary data structures.
   Method: *public static<T> boolean searchStack(Stack<T> stack, T elem)*

```
public static <T> boolean searchStack(Stack<T> stack, T elem) {
     if (stack.empty())
          return false;
     T top = stack.pop();
     boolean result;
     if (top.equals(elem))
          result = true;
     else
          result = searchStack(stack, elem);
     stack.push(top);
     return result;
}
```

Problem 2

1. Write the recursive method ***search*** member of the class ***LinkedList*** that searches for an element *elem* in the list and returns true if it's found or false otherwise. You are not allowed to use any auxiliary data structures or call any of the ***LinkedList*** methods.
   Method: *public boolean search(T elem)*

```
public boolean search(T elem) {
    return recSearch(head, elem);
}

private boolean recSearch(Node<T> pointer, T elem) {
    if (pointer == null)
        return false;
    if (pointer.data.equals(elem))
        return true;
    return recSearch(pointer.next, elem);
}
```

2. Write the static recursive method ***searchList*** that searches for an element *elem* in a List *list* and returns true if it's found or false otherwise. You are not allowed to use any auxiliary data structures.
   Method: *public static<T> boolean searchList(List<T> list, T elem)*

```
public static <T> boolean searchList(List<T> list, T elem) {
    if (list.empty())
        return false;
    list.findFirst();
    return recSearch(list, elem);
}

private static <T> boolean recSearch(List<T> list, T elem) {
    if (list.last())
        return list.retrieve().equals(elem);
    if (list.retrieve().equals(elem))
        return true;
    list.findNext();
    return recSearch(list, elem);
}
```