

**Department of Computer Science,
Data Structures (CSC212),
Tutorial (Week 3)
Lists**

Node	Array List	Linked List
<pre>- data : T - next : Node<T></pre>	<pre>- maxsize : int - size : int - current : int - nodes : T[]</pre>	<pre>- head : Node<T> - current : Node<T></pre>
<pre>+ Node() + Node(t : T) + Setters/getters</pre>	<pre>+ list(n : int) + findFirst():void + findNext():void + retrieve():T + update(e : T):void + insert(e : T):void + remove():void + full():boolean + empty():boolean + last():boolean + first():boolean</pre>	<pre>+ list() + list(t : T) + findFirst():void + findNext():void + retrieve():T + update(e : T):void + insert(e : T):void + remove():void + full():boolean + empty():boolean + last():boolean + first():boolean</pre>

Question 1. (Lists)

- (a) Add a new method/operation to the ADT List (Array Implementation) to find the previous element of the current. The method `findPrevious(l : List)`, is specified below. Test this method in a test class for the ADT List by calling it on a list object.
- (b) Add a new method/operation to the ADT List (LinkedList Implementation) to find the previous element of the current. The method `findPrevious(l : List)`, is specified below. Test this method in a test class for the ADT List by calling it on a list object.
- (c) Implement `findPrevious()` as a static method in the test class. The test class should have the static `main()` method that declares variables of type ADT List and calls `findPrevious()` static method. The specification of the method is as follows:

```
void findPrevious (l : List)
```

Precondition/Requires: The list `l` should not be empty or the current node should not be the first node.

Results/Actions: Sets the current to point to the node previous to the current node.

Question 2. (Lists)

- (a) Add a new method/operation to the ADT List (Array Implementation) to find the element equal element t. The method search(l : List, t : T), is specified below. Test this method in a test class for the ADT List by calling it on a list object.
- (b) Add a new method/operation to the ADT List (LinkedList Implementation) to find the element equal element t. The method search(l : List, t : T), is specified below. Test this method in a test class for the ADT List by calling it on a list object.
- (c) Implement search() as a static method in the test class. The test class should have the static main() method that declares variables of type ADT List and calls search() static method. The specification of the method is as follows:

boolean search (l : List, t : T)

Precondition/Requires: The list l should not be empty.

Results/Actions: Sets the current to point to the node with data equal to t. Return true if found. False otherwise.

Question 3. (Lists)

- (a) Add a new method/operation to the ADT List (Array Implementation) to find the greatest element. The method max(l : List), is specified below. Test this method in a test class for the ADT List by calling it on a list object.
- (b) Add a new method/operation to the ADT List (Array Implementation) to find the greatest element. The method max(l : List), is specified below. Test this method in a test class for the ADT List by calling it on a list object.
- (c) Implement max() as a static method in the test class. The test class should have the static main() method that declares variables of type ADT List and calls max() static method. The specification of the method is as follows:

T max (l : List)

Precondition/Requires: The list l should not be empty.

Results/Actions: Return the Maximum value.