

# Linked List Testing: Concept Challenge

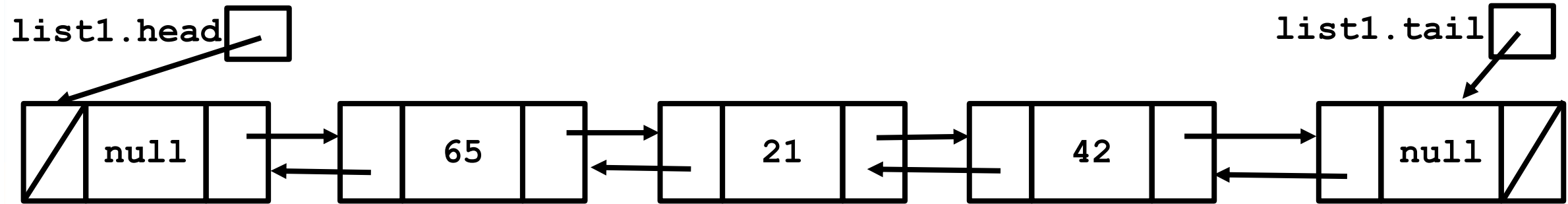


# Concept Challenge: Procedure

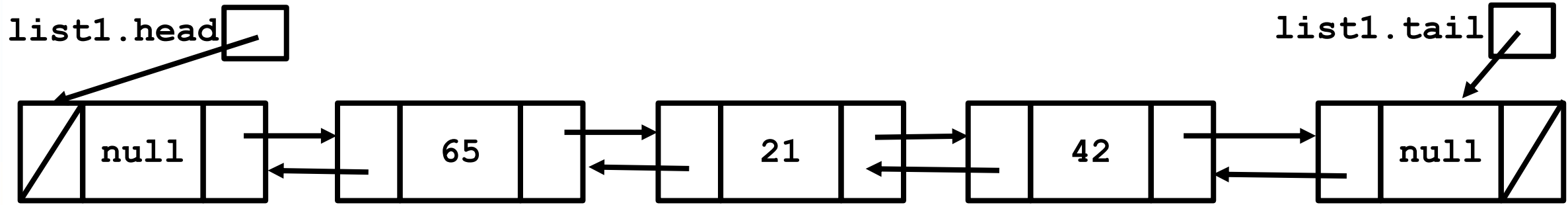
- **Pause** Try to solve the problem yourself
- **Discuss** with other learners (if you can)
- **Watch** the UC San Diego learners video
- **Answer** the question again
- **Confirm** your understanding with our explanation



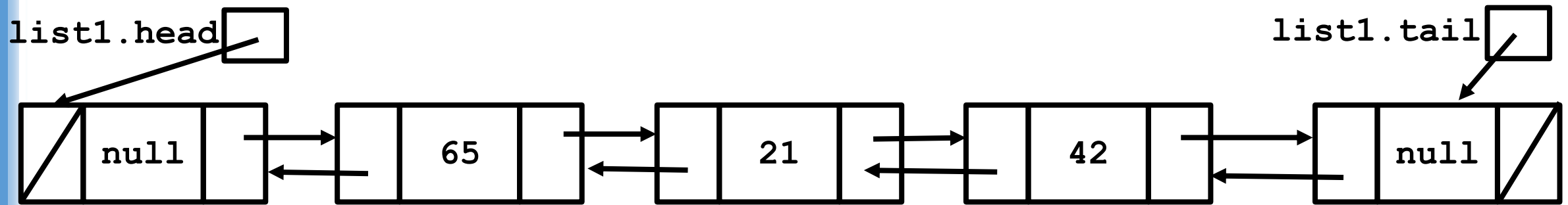
# Testing remove



# Testing remove



```
// removes the element at the index indicated or, if the index
// is invalid, throw and IndexOutOfBoundsException
public E remove(int index)
```



```
// removes the element at the index indicated or, if the index  
// is invalid, throw and IndexOutOfBoundsException  
public E remove(int index)
```

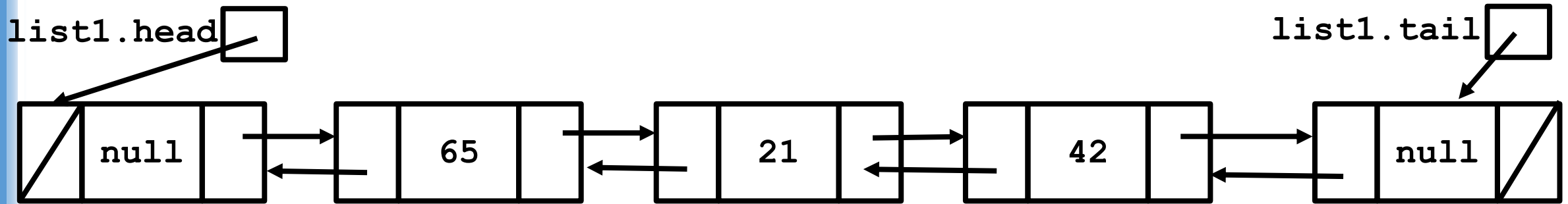
In testRemove() you run:

```
int a = list1.remove(0);
```

**What verification code should you include to make sure this operation worked correctly?**

# Learner video here





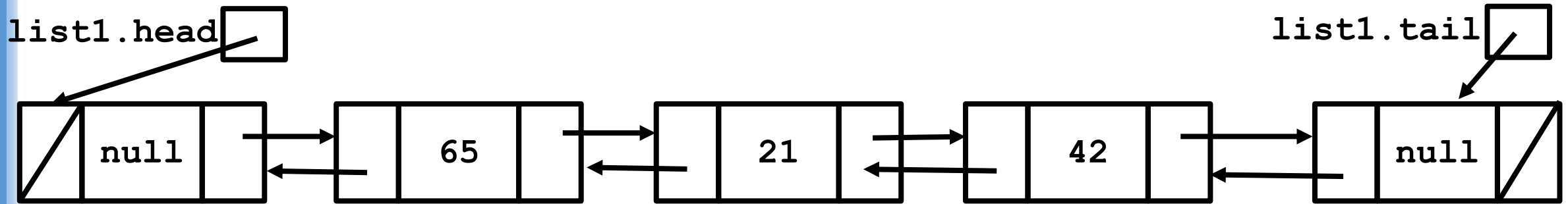
```
// removes the element at the index indicated or, if the index  
// is invalid, throw and IndexOutOfBoundsException  
public E remove(int index)
```

In testRemove() you run:

```
int a = list1.remove(0);
```

**What verification code should you include to make sure this operation worked correctly?**

✓ **Verify that a has the value 65**



```
// removes the element at the index indicated or, if the index  
// is invalid, throw and IndexOutOfBoundsException  
public E remove(int index)
```

In testRemove() you run:

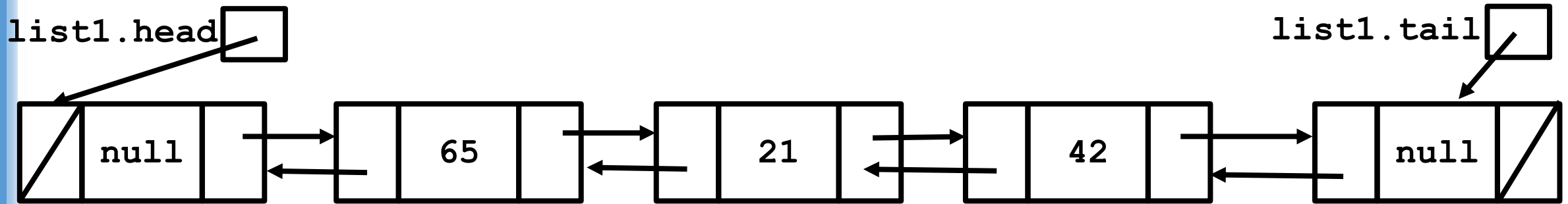
```
int a = list1.remove(0);
```

**What verification code should you include to make sure this operation worked correctly?**

✓ Verify that a has the value 65

✗ **Call list1.get(-1) to check if it throws a IndexOutOfBoundsException**





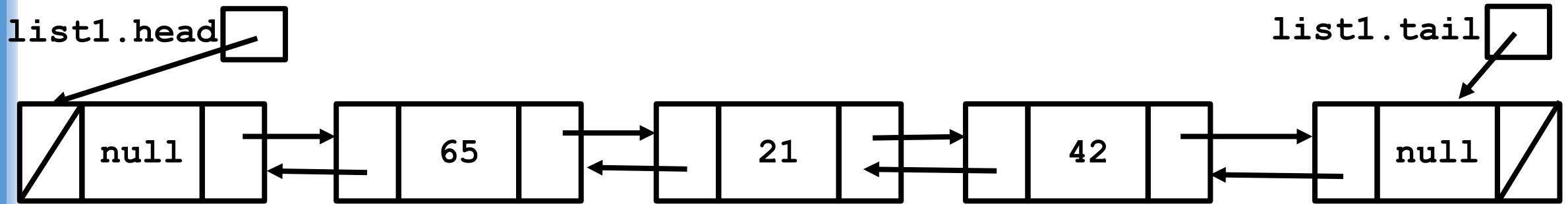
```
// removes the element at the index indicated or, if the index  
// is invalid, throw and IndexOutOfBoundsException  
public E remove(int index)
```

In testRemove() you run:

```
int a = list1.remove(0);
```

**What verification code should you include to make sure this operation worked correctly?**

- ✓ Verify that a has the value 65
- ✗ ~~Call list1.get(-1) to check if it throws a IndexOutOfBoundsException~~
- ✓ **Call list1.get(0) and check that index 0 has the value 21**



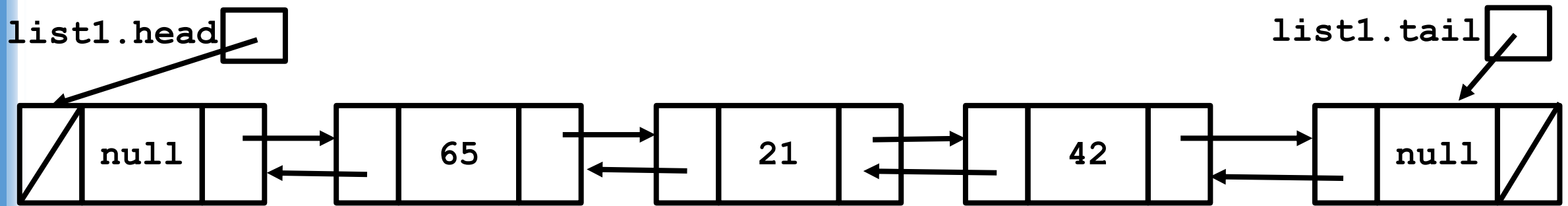
```
// removes the element at the index indicated or, if the index  
// is invalid, throw and IndexOutOfBoundsException  
public E remove(int index)
```

In testRemove() you run:

```
int a = list1.remove(0);
```

**What verification code should you include to make sure this operation worked correctly?**

- ✓ Verify that a has the value 65
- ✗ ~~Call list1.get(-1) to check if it throws a IndexOutOfBoundsException~~
- ✓ Call list1.get(0) and check that index 0 has the value 21
- ✗ **Call list1.get(1) and check that index 1 has the value 42**



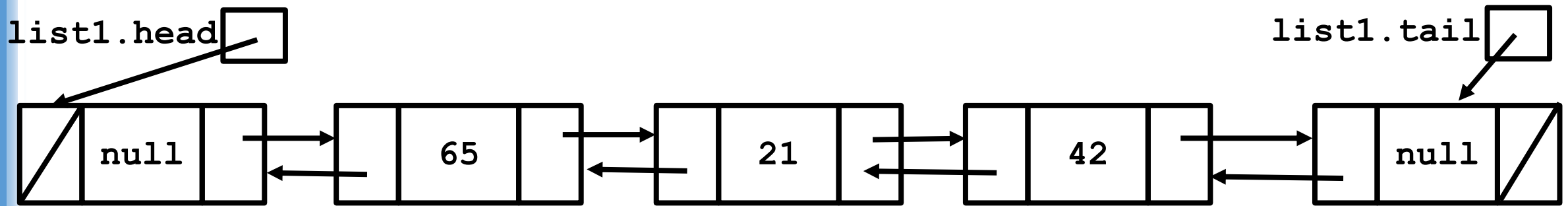
```
// removes the element at the index indicated or, if the index  
// is invalid, throw and IndexOutOfBoundsException  
public E remove(int index)
```

In testRemove() you run:

```
int a = list1.remove(0);
```

**What verification code should you include to make sure this operation worked correctly?**

- ✓ Verify that a has the value 65
- ✗ ~~Call list1.get(-1) to check if it throws a IndexOutOfBoundsException~~
- ✓ Call list1.get(0) and check that index 0 has the value 21
- ✗ ~~Call list1.get(1) and check that index 1 has the value 42~~
- ✗ **Call list1.get(2) to check if it throws a NullPointerException**



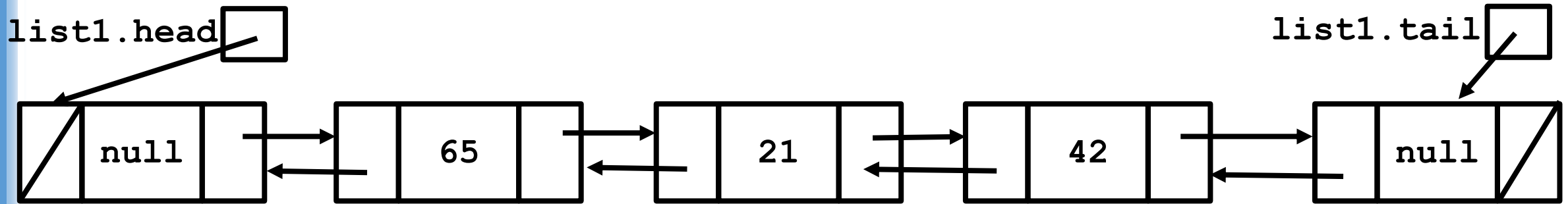
```
// removes the element at the index indicated or, if the index  
// is invalid, throw and IndexOutOfBoundsException  
public E remove(int index)
```

In testRemove() you run:

```
int a = list1.remove(0);
```

**What verification code should you include to make sure this operation worked correctly?**

- ✓ Verify that a has the value 65
- ✗ ~~Call list1.get(-1) to check if it throws a IndexOutOfBoundsException~~
- ✓ Call list1.get(0) and check that index 0 has the value 21
- ✗ ~~Call list1.get(1) and check that index 1 has the value 42~~
- ✗ ~~Call list1.get(2) to check if it throws a NullPointerException~~
- ✓ **Call list1.size() to check if size is 2**



```
// removes the element at the index indicated or, if the index  
// is invalid, throw and IndexOutOfBoundsException  
public E remove(int index)
```

In `testRemove()` you run:

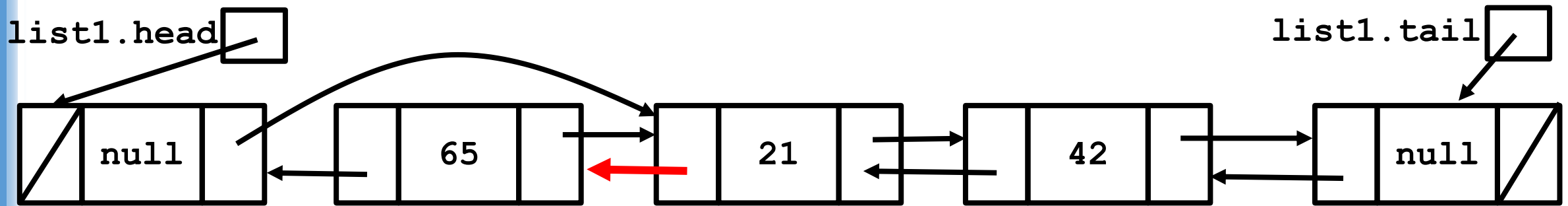
```
int a = list1.remove(0);
```

**What verification code should you include to make sure this operation worked correctly?**

Verify that `a` has the value 65

Call `list1.get(0)` and check that index 0 has the value 21

Call `list1.size()` to check if size is 2



```
// removes the element at the index indicated or, if the index  
// is invalid, throw and IndexOutOfBoundsException  
public E remove(int index)
```

In `testRemove()` you run:

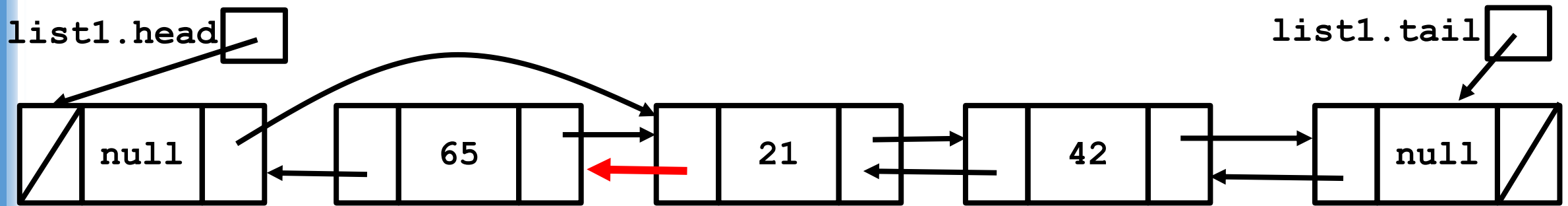
```
int a = list1.remove(0);
```

**What about this bug?**

Verify that `a` has the value 65

Call `list1.get(0)` and check that index 0 has the value 21

Call `list1.size()` to check if size is 2



```
// removes the element at the index indicated or, if the index  
// is invalid, throw and IndexOutOfBoundsException  
public E remove(int index)
```

In `testRemove()` you run:

```
int a = list1.remove(0);
```

**What about this bug?**

Verify that `a` has the value 65

Call `list1.get(0)` and check that index 0 has the value 21

Call `list1.size()` to check if size is 2

**Check that `list1.get(0).prev` is equal to `list1.head`**

**breaks black box testing,  
but you can include if you want**