

More linked lists

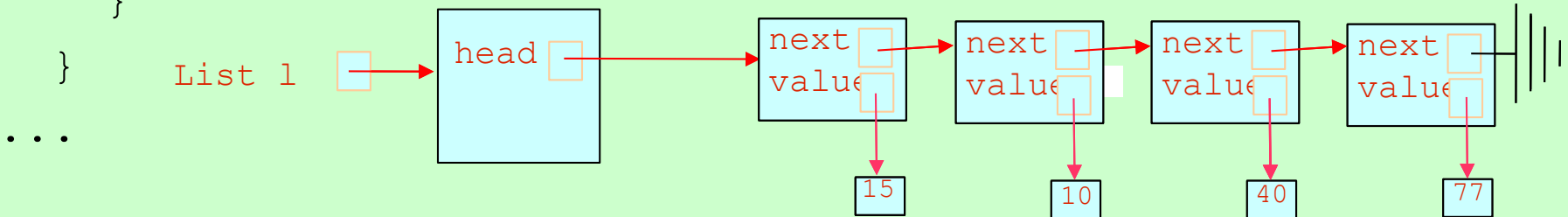
4/8/26

Administrivia

- HW 2 (test cases for Book) due tonight (4/8)
- Extra credit survey about parallel and distributed computing due tonight

Recall: LinkedList

```
public class OurLinkedList<T> implements OurList<T> {  
    private Node head;  
    //optionally: could have size  
  
    public class Node {  
        public T value;  
        public Node next;  
  
        public Node(T value, Node next) {  
            this.value = value;  
            this.next = next;  
        }  
    }  
}
```



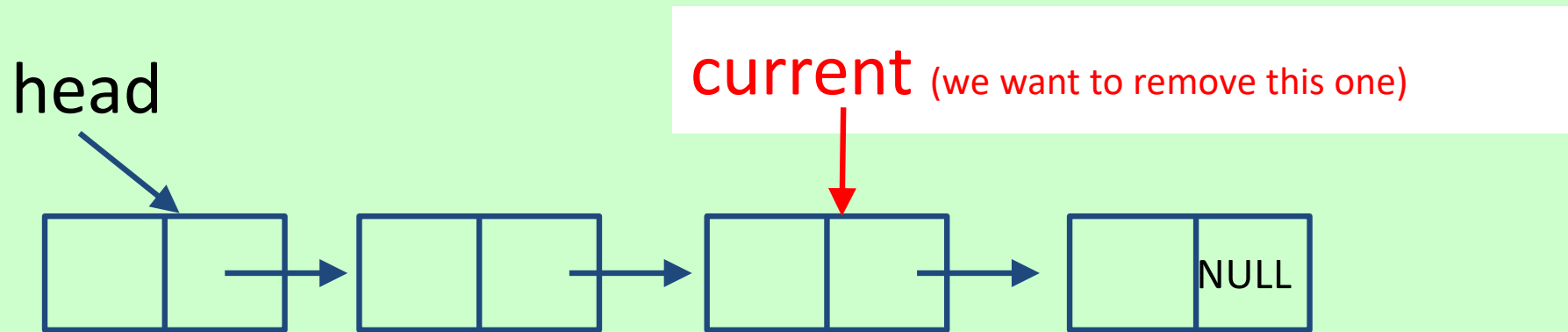
Write get method

(returns value at given index or throws `IndexOutOfBoundsException`)

```
public T get(int index) {
```

```
    ...
```

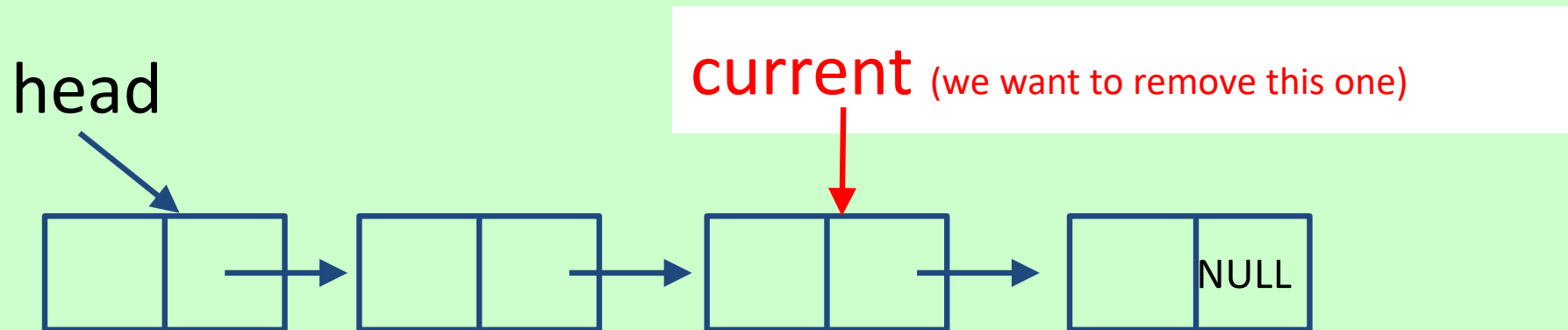
Suppose we have a **reference (current)** to the node containing the item to be removed.



What additional information do we need to successfully remove the node?

- A) Nothing additional.
- B) A reference to the node immediately prior to the deleted node.
- C) A reference to the node immediately after the node to be deleted.
- D) Both B and C.

Suppose we have a **reference (current)** to the node containing the item to be removed.



What additional information do we need to successfully remove the node?

- A) Nothing additional.
- B) A reference to the node immediately prior to the deleted node.
- C) A reference to the node immediately after the node to be deleted.
- D) Both B and C.

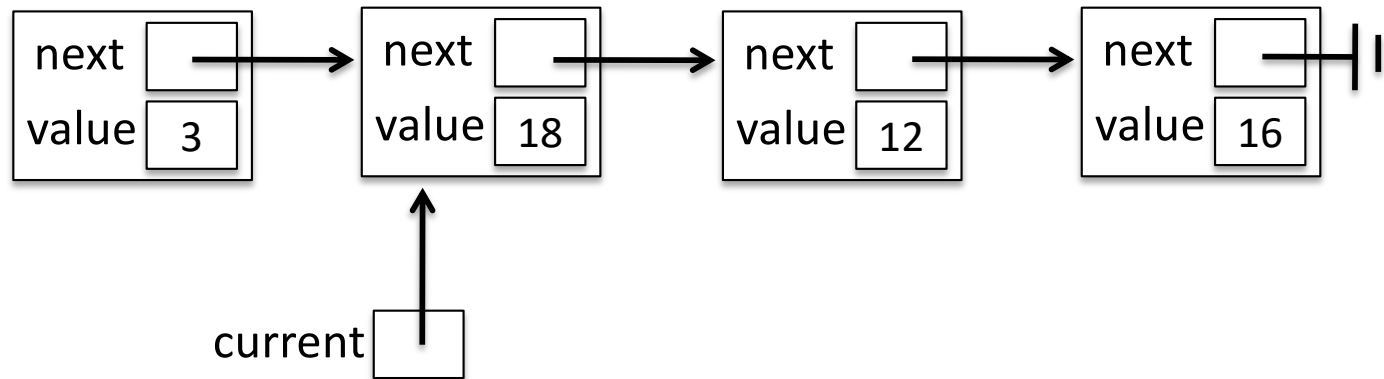
Basic idea

- method remove(T)
 - curr: reference to “current” Node
used to find Node to remove
 - prev: reference to Node before the current one

Basic idea

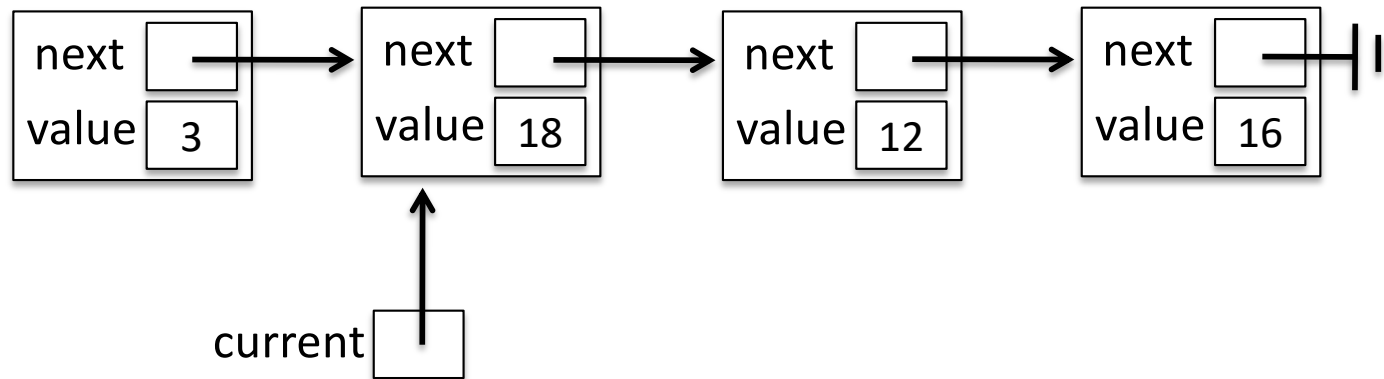
- method `remove(T)`
 - `curr`: reference to “current” Node
used to find Node to remove
 - `prev`: reference to Node before the current one
- Alternate idea: just use one reference and stop on Node before the one to remove

For the situation given below, which of the following is the value to which `current.next.value` evaluates?



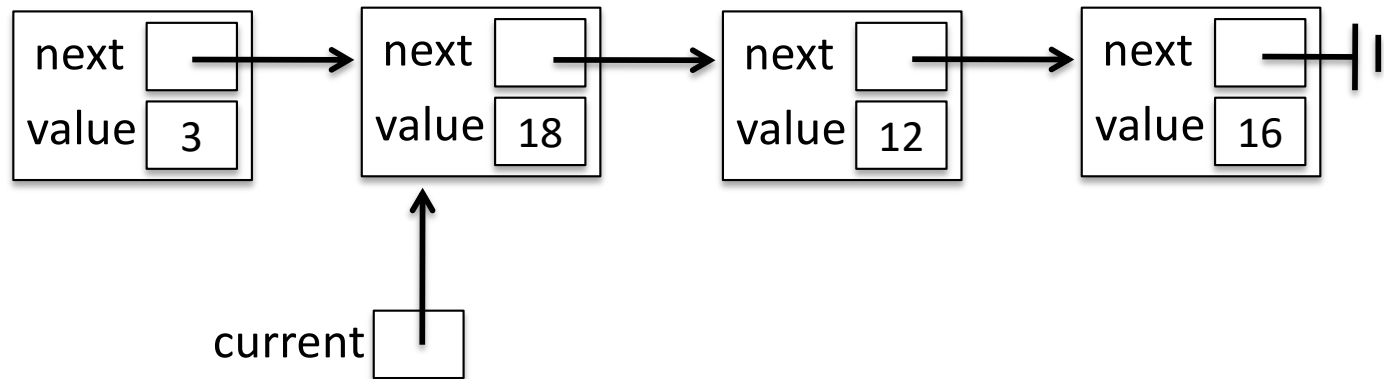
- A. 3
- B. 18
- C. 12
- D. 16
- E. None of the above

For the situation given below, which of the following is the value to which `current.next.value` evaluates?



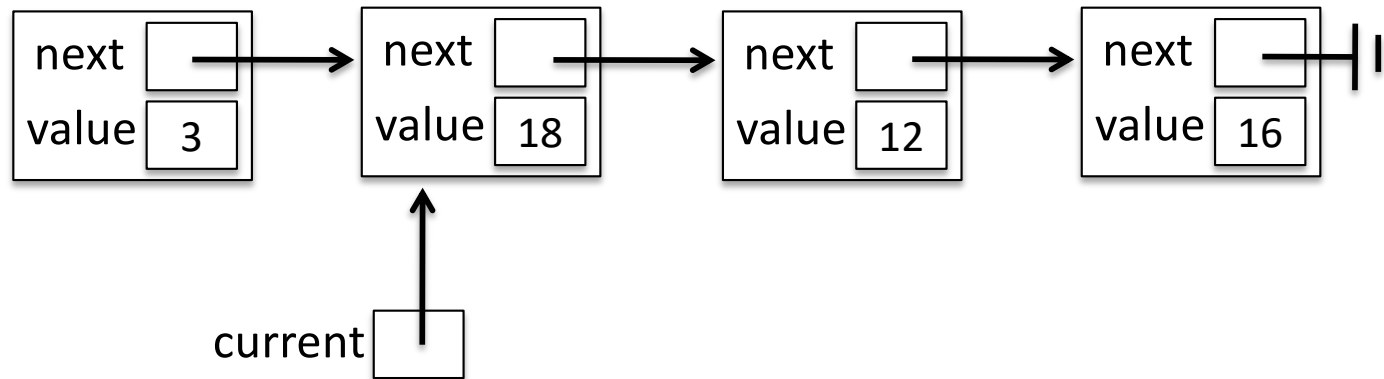
- A. 3
- B. 18
- C. 12
- D. 16
- E. None of the above

For the situation given below, which of the following is the value to which `current.next.next.value` evaluates?



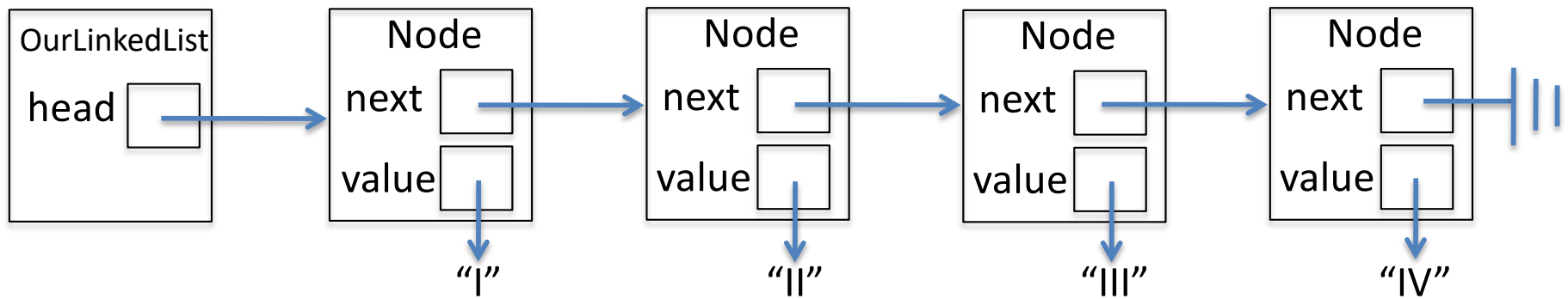
- A. 3
- B. 18
- C. 12
- D. 16
- E. None of the above

For the situation given below, which of the following is the value to which `current.next.next.value` evaluates?



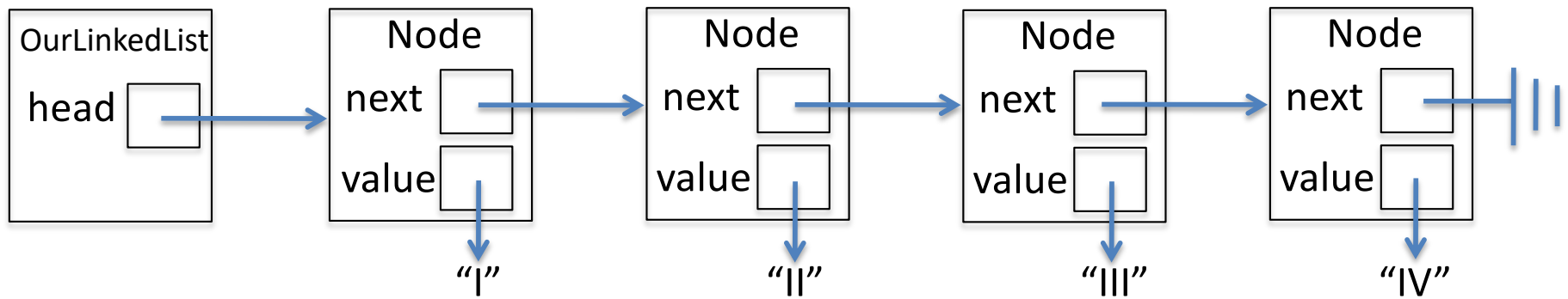
- A. 3
- B. 18
- C. 12
- D. 16**
- E. None of the above

Which of the following is equal to `head.next.value` when called from the `OurLinkedList`?



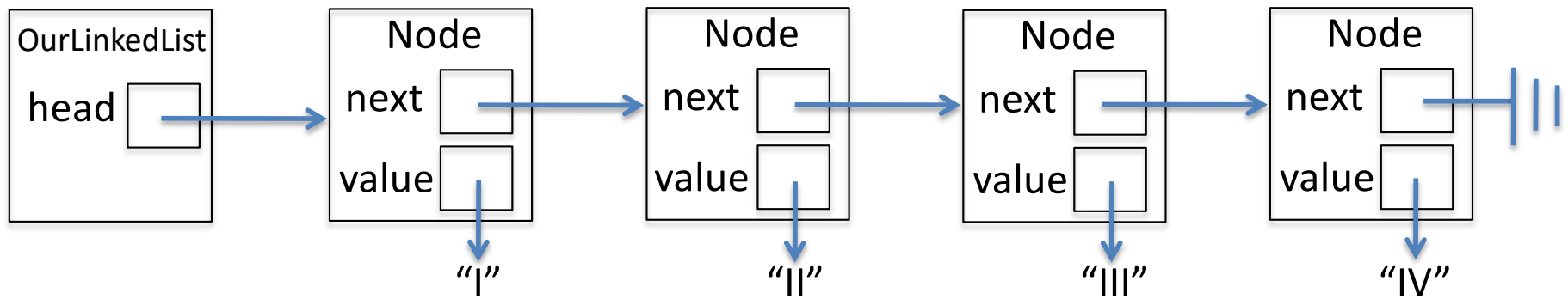
- A. "I"
- B. "II"
- C. "III"
- D. "IV"
- E. Not exactly one of the above

Which of the following is equal to `head.next.value` when called from the `OurLinkedList`?



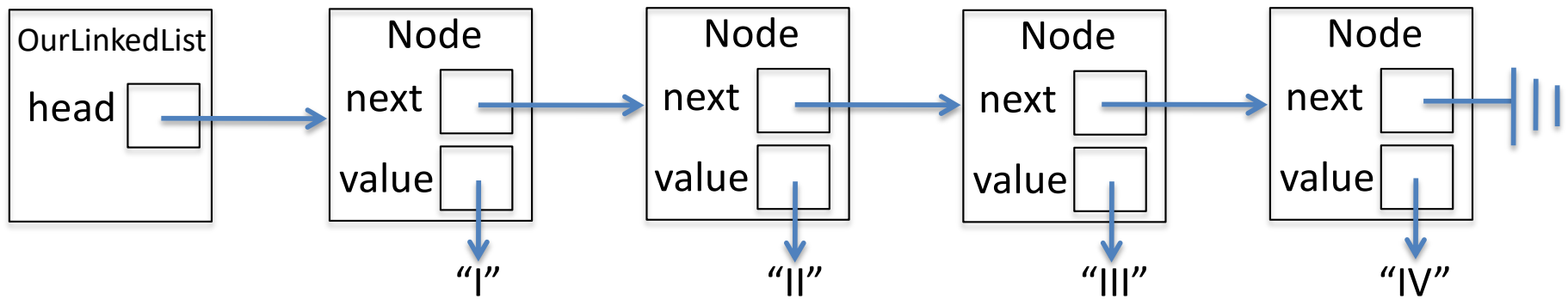
- A. "I"
- B. "II"
- C. "III"
- D. "IV"
- E. Not exactly one of the above

Which of the following removes the Node with “II” from the list when called from OurLinkedList?



- A. `head = head.next;`
- B. `head.next = head;`
- C. `head.next = head.next.next;`
- D. `head.next.next = head.next.next.next;`
- E. Not exactly one of the above

Which of the following removes the Node with “II” from the list when called from OurLinkedList?



- A. `head = head.next;`
- B. `head.next = head;`
- C. `head.next = head.next.next;`
- D. `head.next.next = head.next.next.next;`
- E. Not exactly one of the above