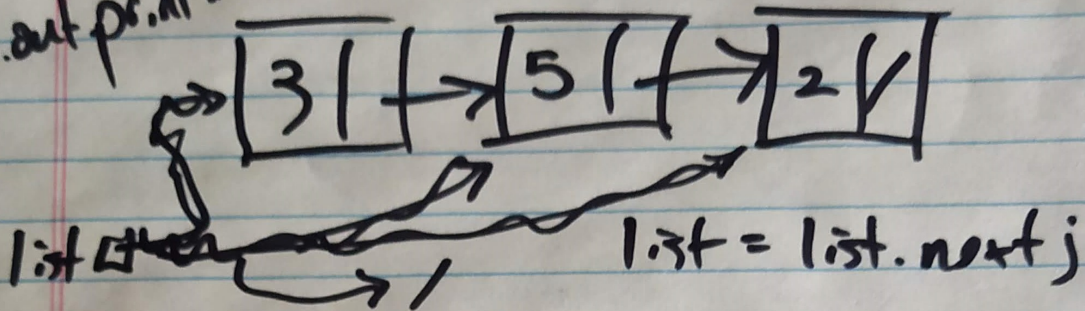


front → [17/1]

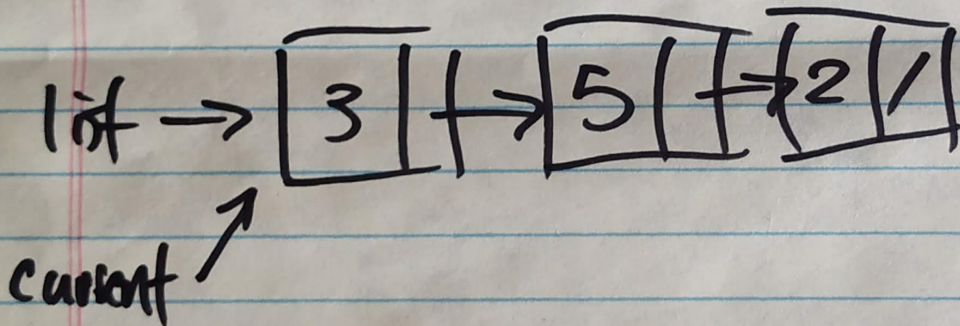
```
public void add (int value) {  
    if (front == null) {  
        front = new ListNode (value);  
    } else {  
        ListNode current = front;  
        while (current.next != null) {  
            current = current.next;  
        }  
        current.next = new ListNode (value);  
    }  
}
```

~

System.out.println(list.data)



```
ListNode list;  
ListNode current = list;
```



```
int i = 0;  
while (current != null) {  
    System.out.println(current.data);  
    current = current.next;  
}
```

```
for (int i = 0; i < arr.length; i++) {  
    System.out.println(arr[i]);  
}
```



```
public class ListNode {
    public int data;          // data stored in this node
    public ListNode next;    // link to next node in the list

    // post: constructs a node with data 0 and null link
    public ListNode() {
        this(0, null);
    }

    // post: constructs a node with given data and null link
    public ListNode(int data) {
        this(data, null);
    }

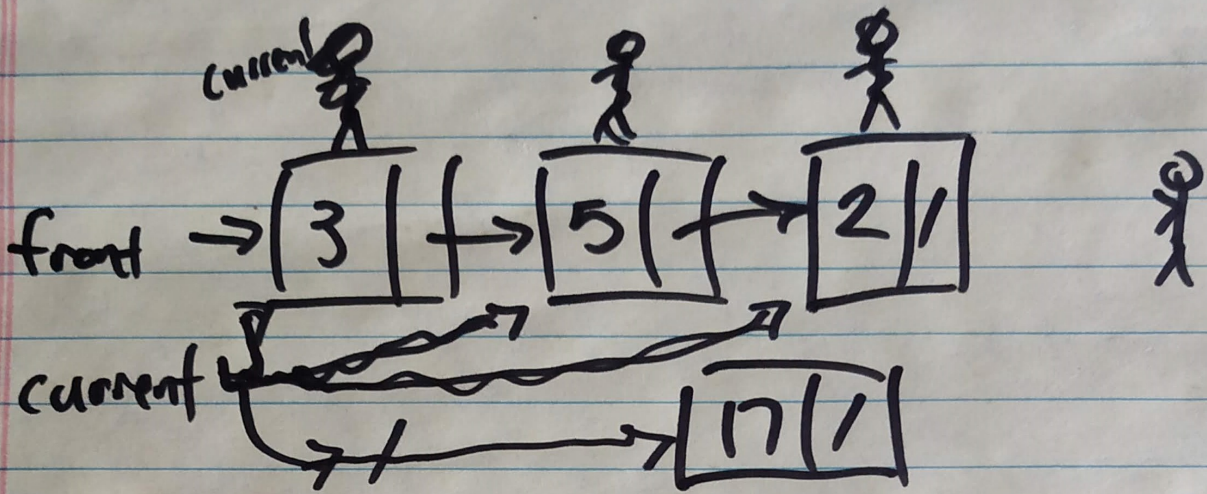
    // post: constructs a node with given data and given link
    public ListNode(int data, ListNode next) {
        this.data = data;
        this.next = next;
    }
}
```

```
public class LinkedIntList {
    private ListNode front;

    public void add(int value) {
        ...
    }
}
```

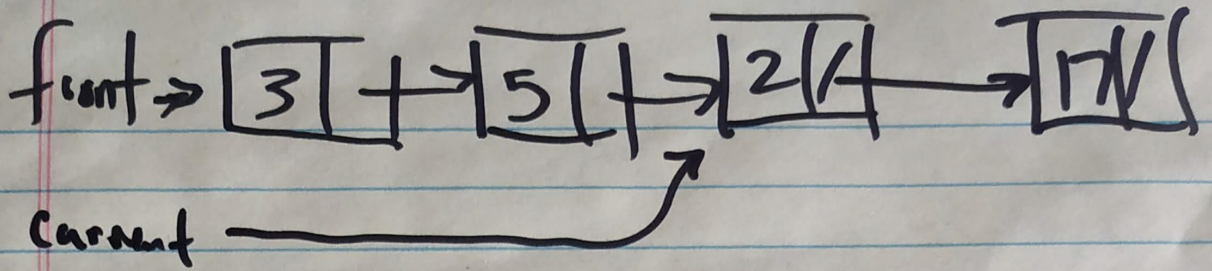
3


```
LinkedList list = new LinkedList();  
list.add(17);
```



```
public void add(int value) {  
    ListNode current = front;  
    while (current != null) {  
        current = current.next;  
    }  
    current = new ListNode(value);  
}
```

"stopping one early"



Ways to change a list:

- change the front field
- change a .next reference

```
public void add (int value) {
    ListNode current = front;
    while (current.next != null) {
        current = current.next;
    }
```

```
    i current.next = new ListNode(value);
```

```
    }
    i current = new ListNode(value);
```

front → 1