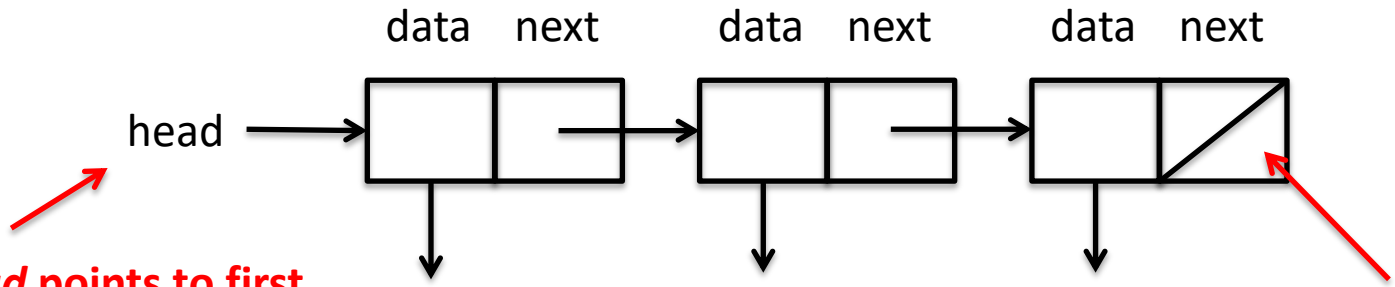
Singly linked list review: elements have data and a next pointer

Singly linked list

"Box-and-pointer" diagram

- **Data in Box**
- **Pointer to next item in List**



***head* points to first item (index 0)**

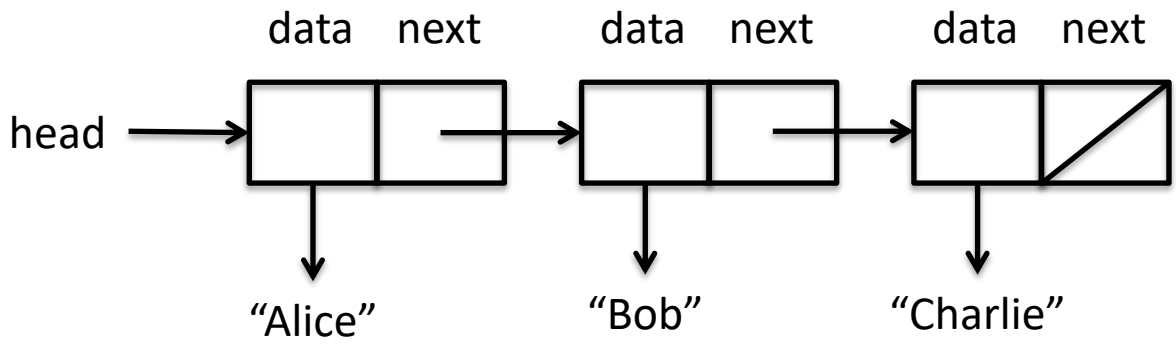
***null* if list is empty**

Slash indicates end of List

next* pointer is *null

To get an item at index i, start at head and march down

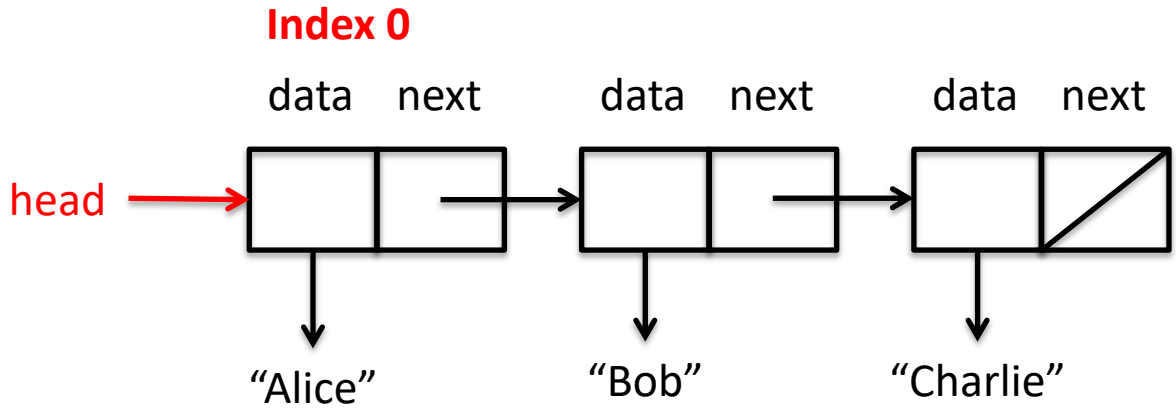
get(i) – return item at specified index



Get item at index 2

To get an item at index i, start at head and march down

get(i) – return item at specified index

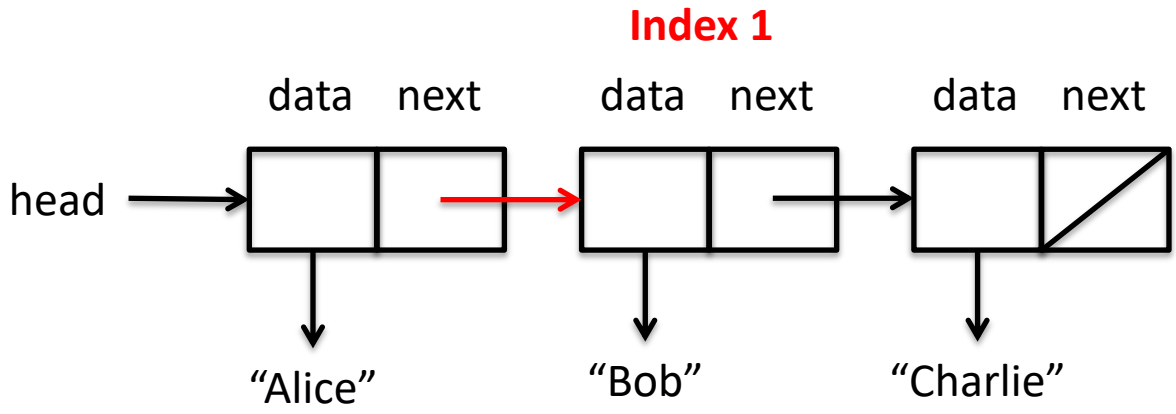


Get item at index 2

1. **Start at head (index 0)**

To get an item at index i, start at head and march down

get(i) – return item at specified index

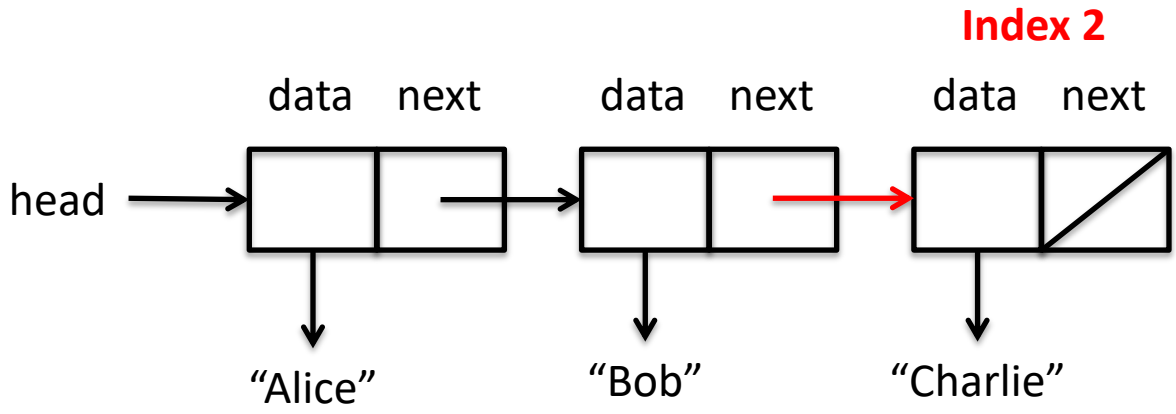


Get item at index 2

1. **Start at head (index 0)**
2. **Follow next pointer to index 1**

To get an item at index i, start at head and march down

get(i) – return item at specified index

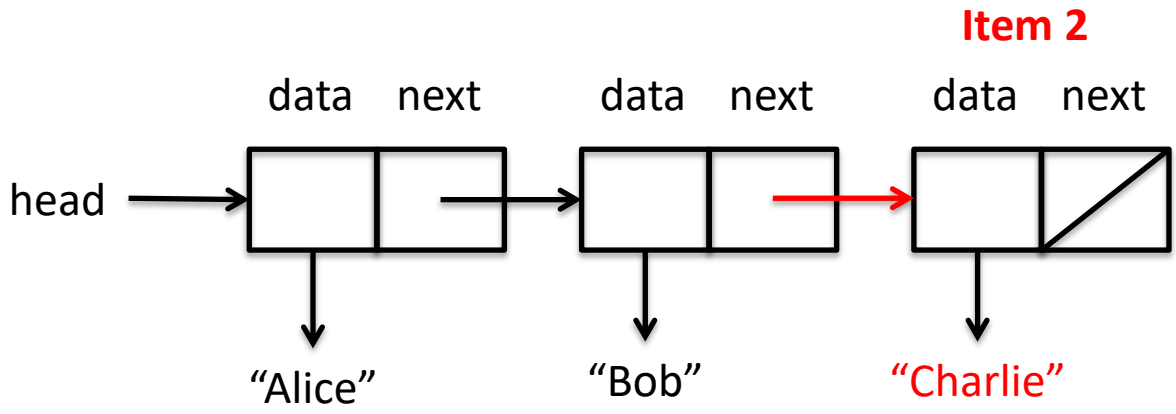


Get item at index 2

1. **Start at head (index 0)**
2. **Follow next pointer to index 1**
3. **Follow next pointer to index 2**

To get an item at index i, start at head and march down

get(i) – return item at specified index

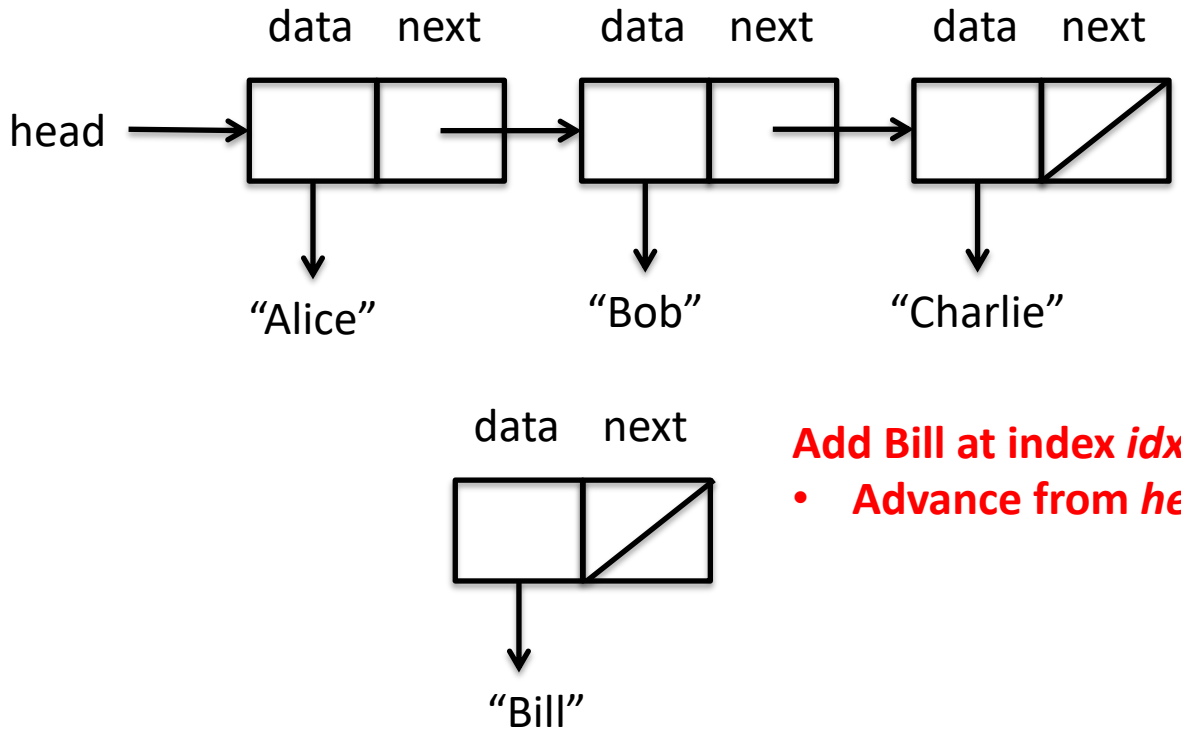


Get item at index 2

1. **Start at head (index 0)**
2. **Follow next pointer to index 1**
3. **Follow next pointer to index 2**
4. **Return "Charlie" (index 2)**

add() "splices in" a new object anywhere in the list by updating next pointers

add(1, "Bill")

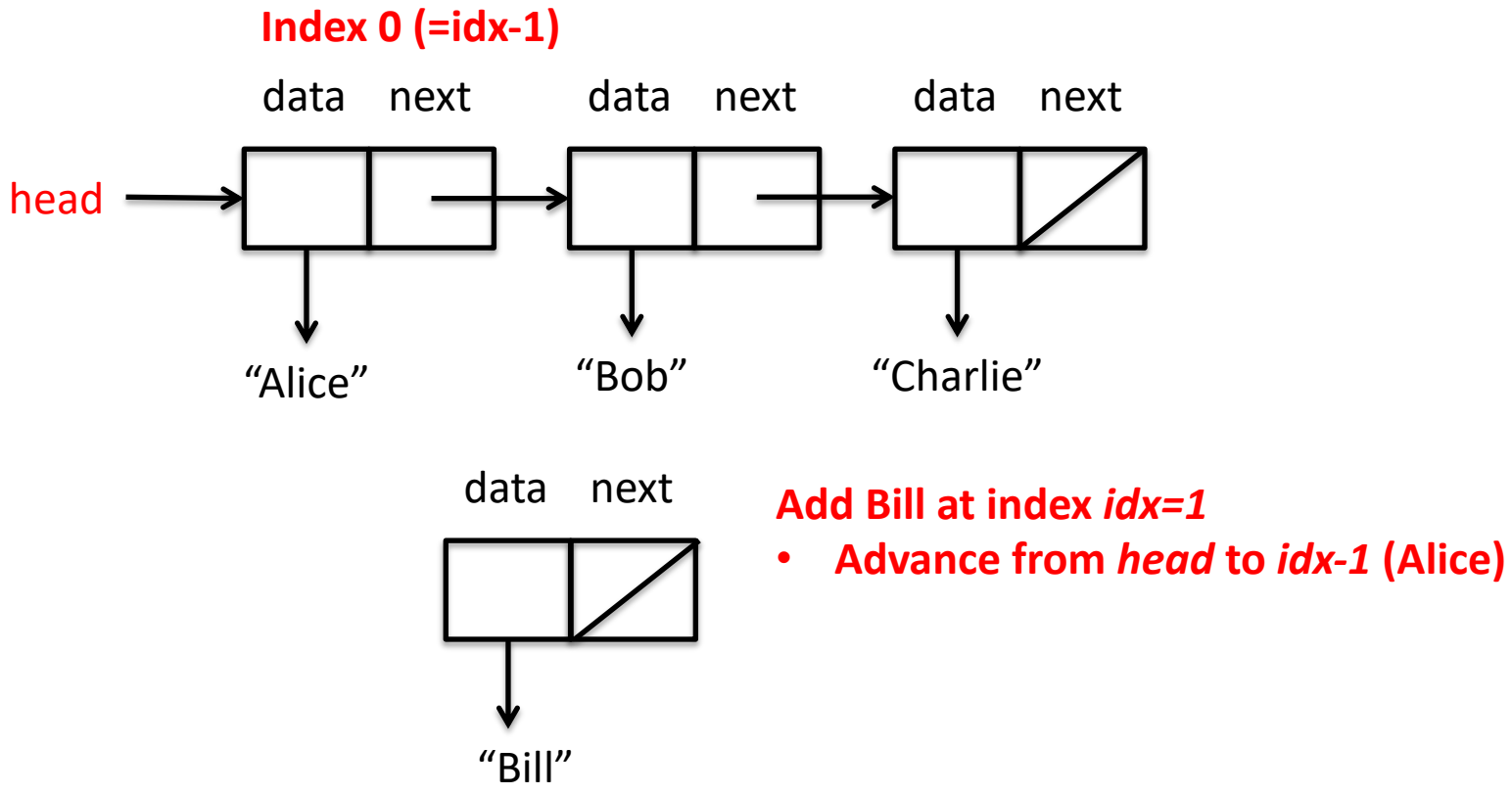


Add Bill at index *idx=1*

- **Advance from *head* to *idx-1* (Alice)**

add() "splices in" a new object anywhere in the list by updating next pointers

add(1, "Bill")

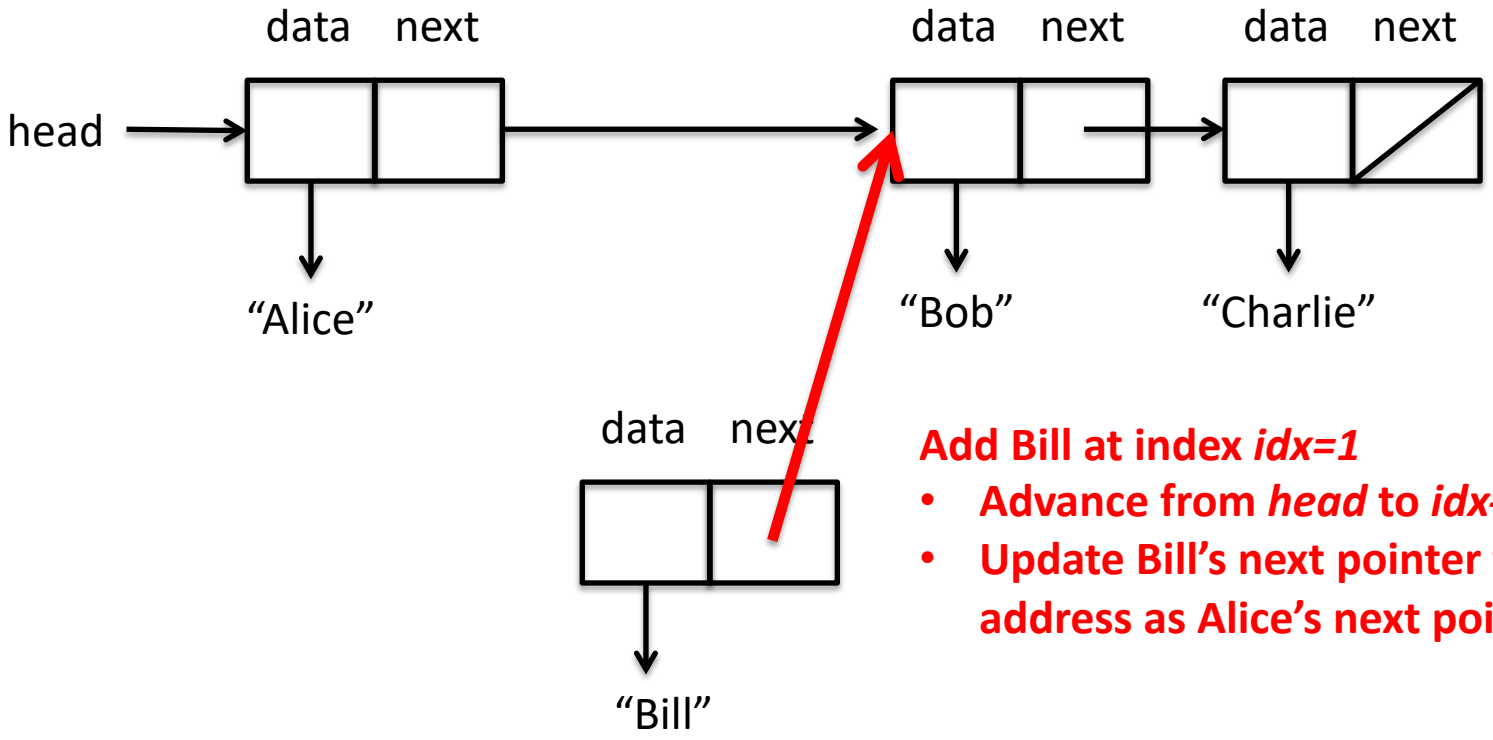


Add Bill at index *idx=1*

- **Advance from *head* to *idx-1* (Alice)**

add() "splices in" a new object anywhere in the list by updating next pointers

add(1, "Bill")

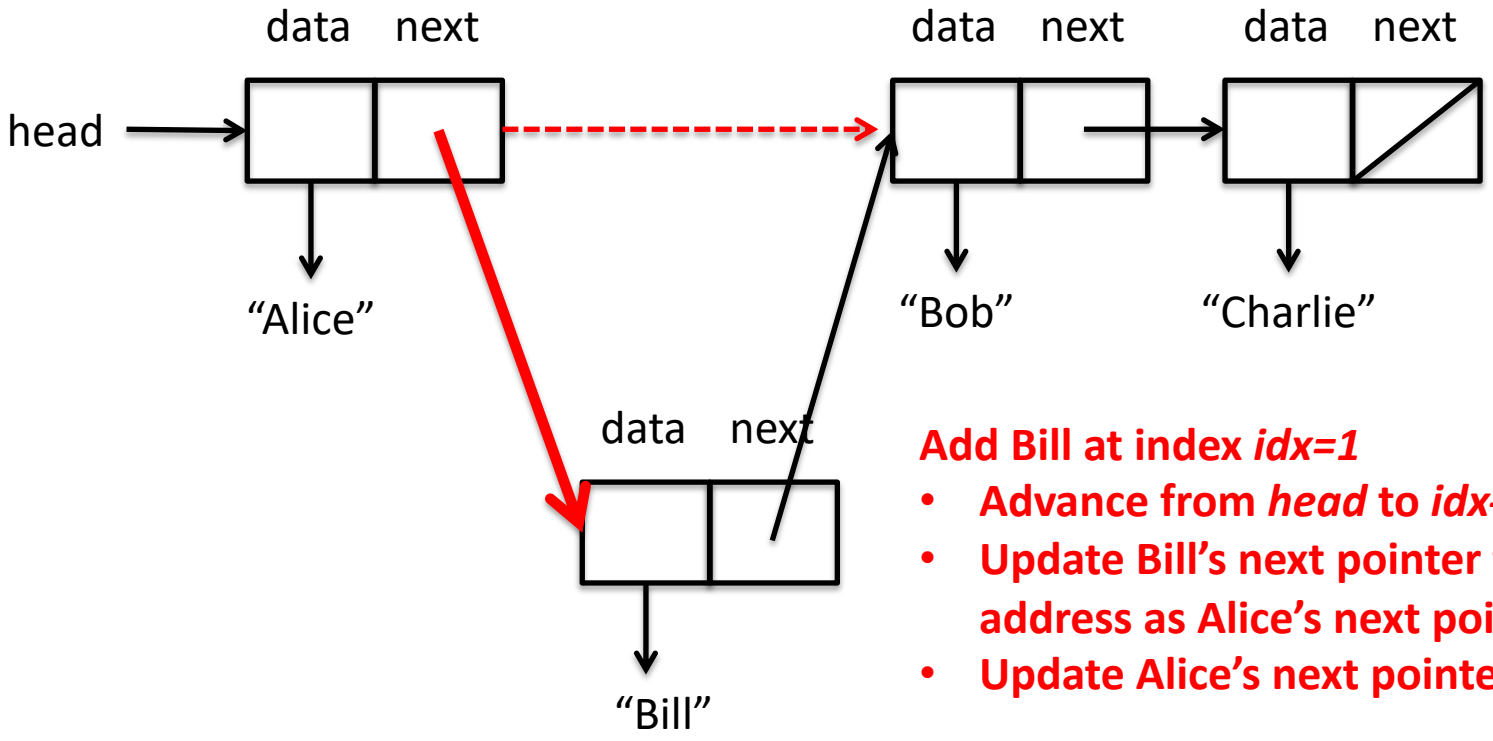


Add Bill at index *idx=1*

- **Advance from *head* to *idx-1* (Alice)**
- **Update Bill's next pointer to same address as Alice's next pointer**

add() "splices in" a new object anywhere in the list by updating next pointers

add(1, "Bill")

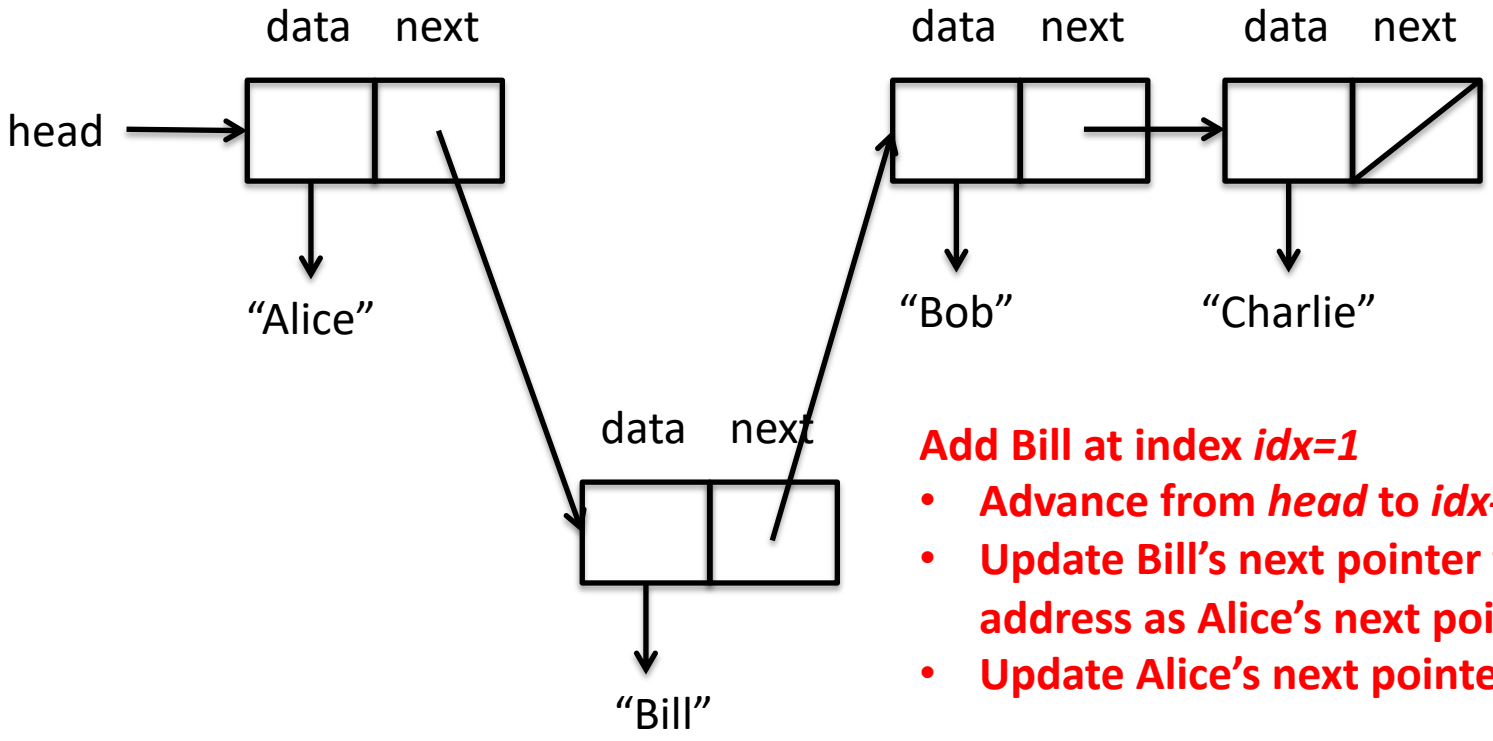


Add Bill at index *idx=1*

- **Advance from *head* to *idx-1* (Alice)**
- **Update Bill's next pointer to same address as Alice's next pointer**
- **Update Alice's next pointer to Bill**

add() "splices in" a new object anywhere in the list by updating next pointers

add(1, "Bill")

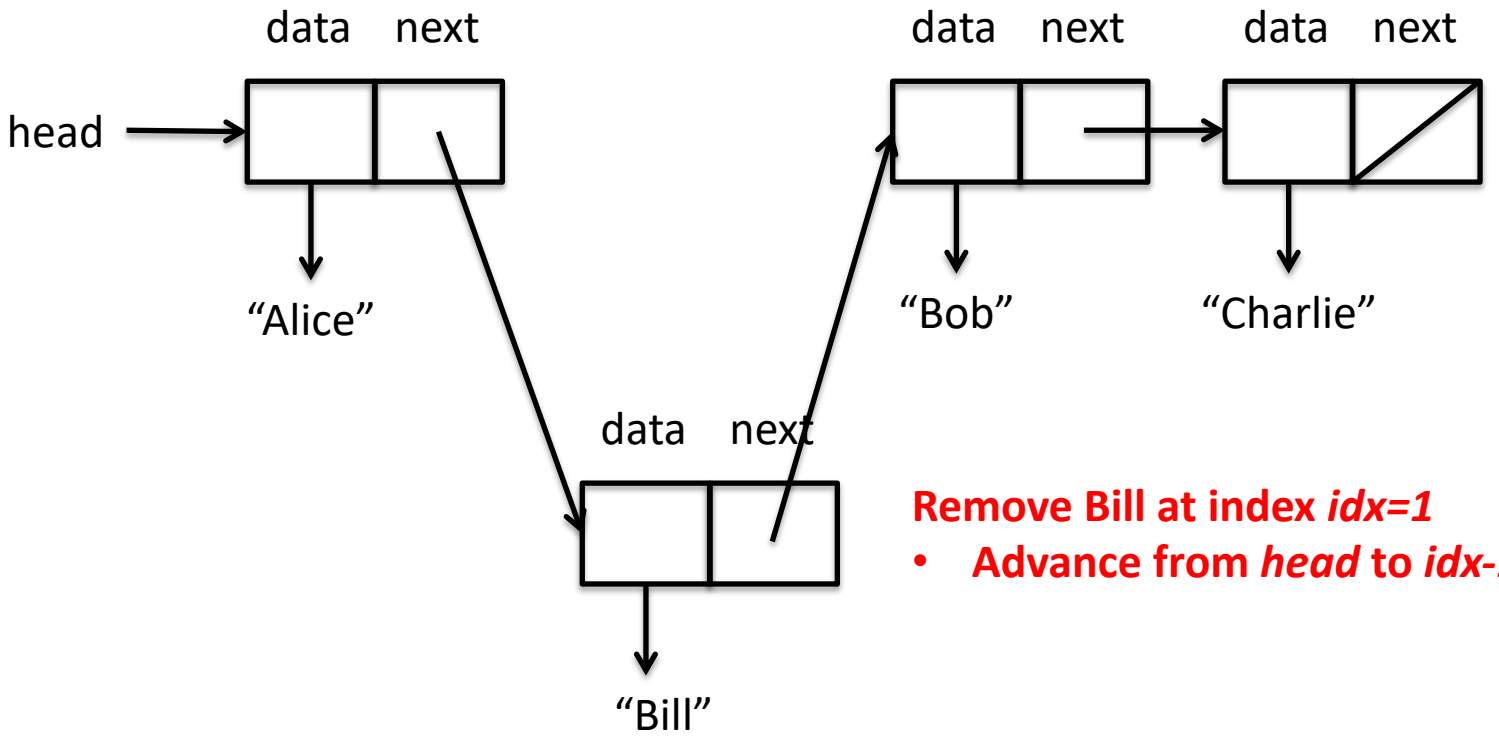


Add Bill at index *idx=1*

- **Advance from *head* to *idx-1* (Alice)**
- **Update Bill's next pointer to same address as Alice's next pointer**
- **Update Alice's next pointer to Bill**

remove() takes an item out of the list by updating next pointer

remove(1)

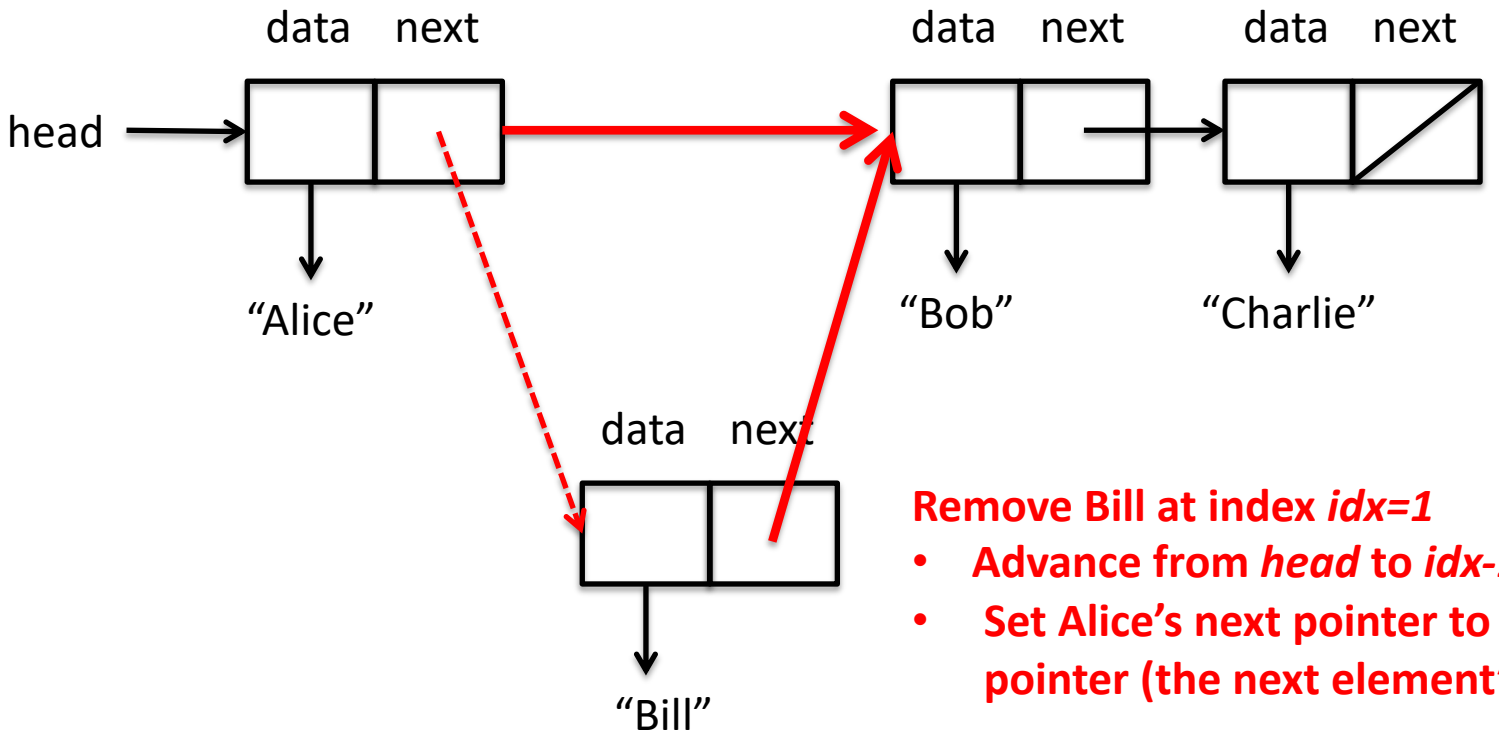


Remove Bill at index *idx=1*

- **Advance from *head* to *idx-1* (Alice)**

remove() takes an item out of the list by updating next pointer

remove(1)

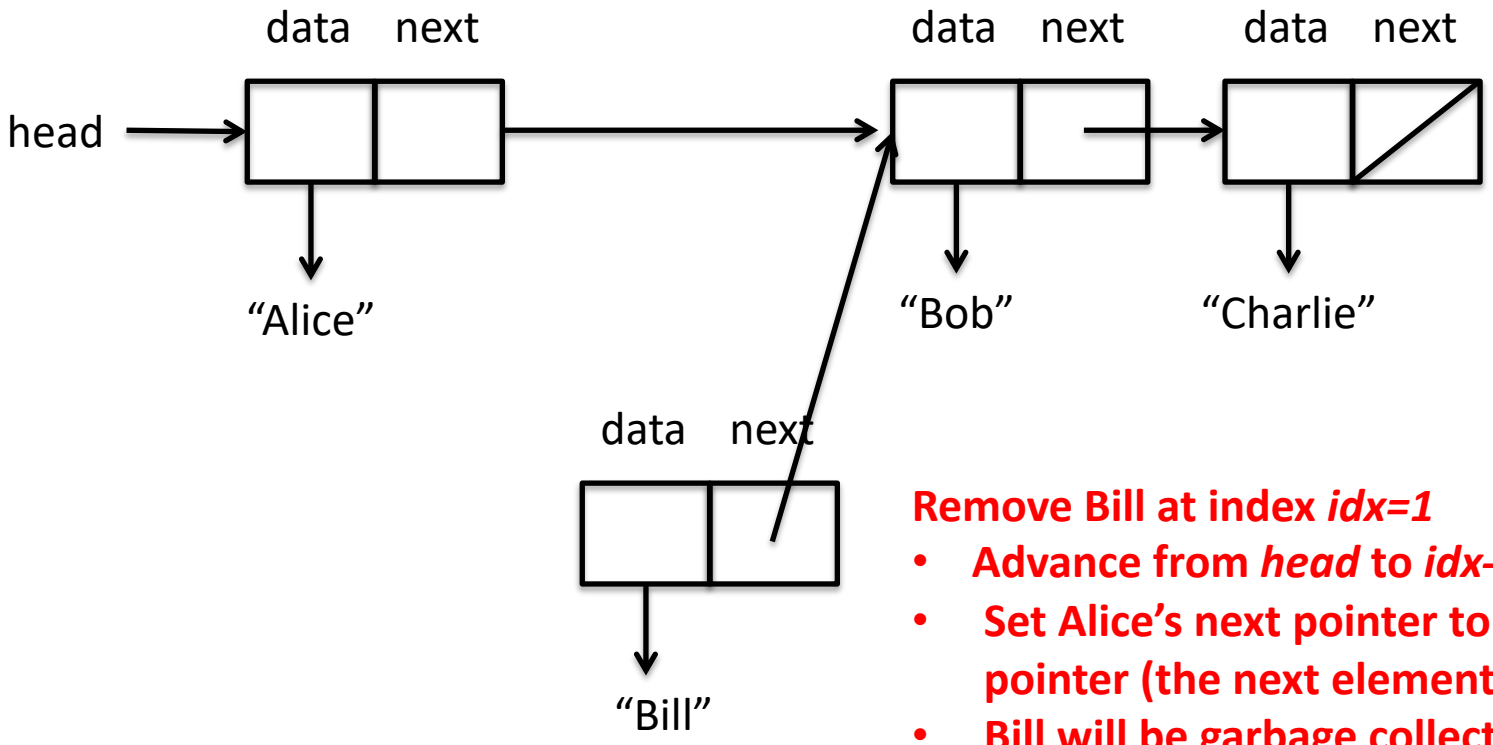


Remove Bill at index *idx=1*

- **Advance from *head* to *idx-1* (Alice)**
- **Set Alice's next pointer to Bill's next pointer (the next element's next)**

remove() takes an item out of the list by updating next pointer

remove(1)



Remove Bill at index *idx=1*

- **Advance from *head* to *idx-1* (Alice)**
- **Set Alice's next pointer to Bill's next pointer (the next element's next)**
- **Bill will be garbage collected (in C we have to call *free()*)**