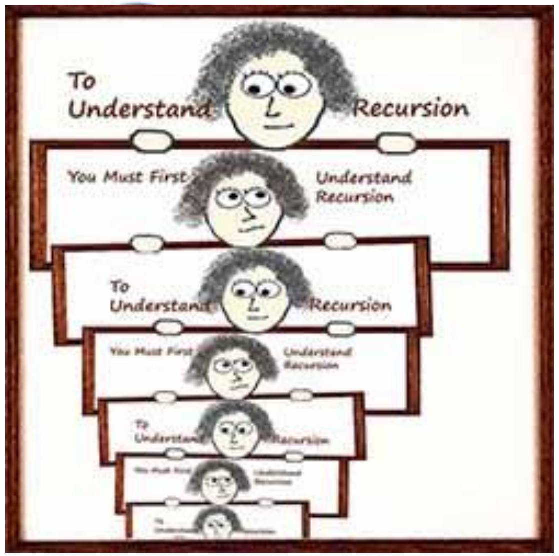
RECURSION AND LINKED-LISTS

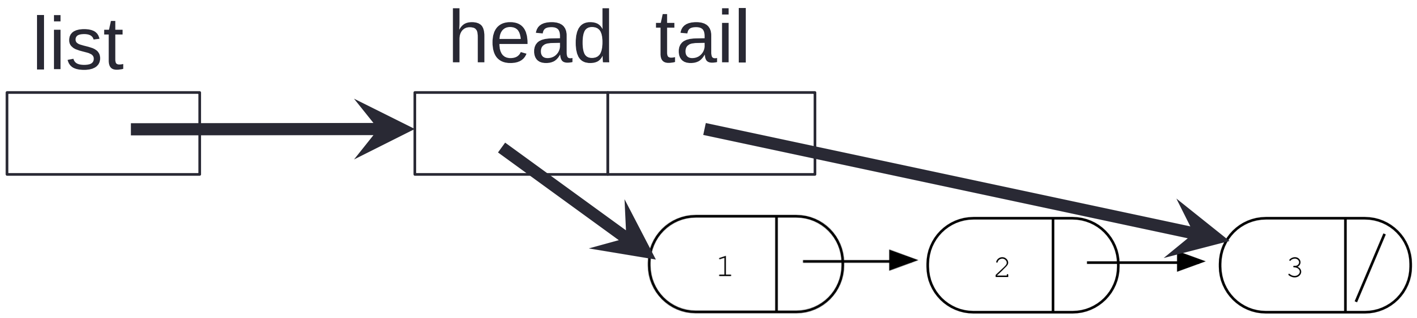
Problem Solving with Computers-I

https://ucsb-cs16-sp17.github.io/


To Understand Recursion
You Must First Understand Recursion
To Understand Recursion
You Must First Understand Recursion
To Understand Recursion
You Must First Understand Recursion

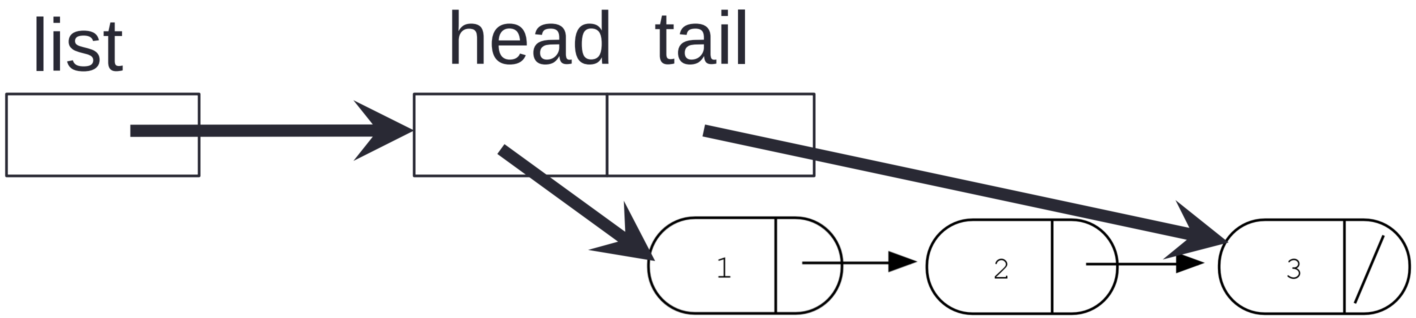
Iterating through the list

int lengthOfList(LinkedList * list)

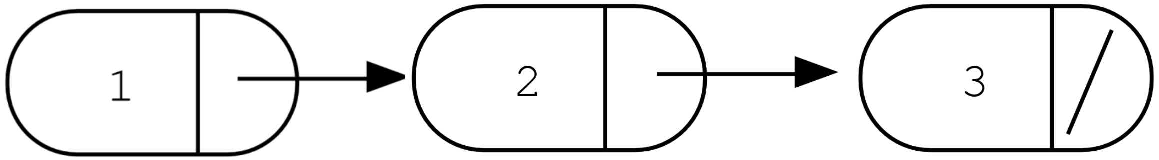


Iterating through the list

int lengthOfListRecursive(LinkedList * list)



Recursion on lists: compute the sum of all elements



```
int sum(Node *head){

    return head->data+sum(head);
}
```

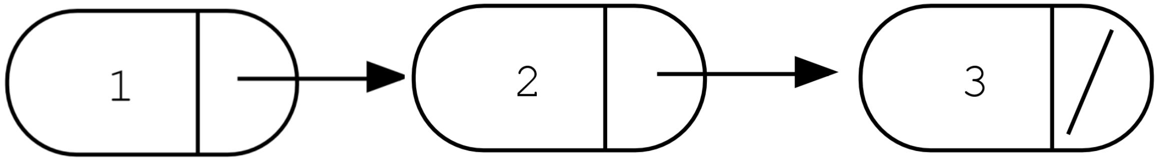
Which of the following is true about the given implementation?

A. It is correct

B. It will not return the correct sum

C. It will result in a segfault

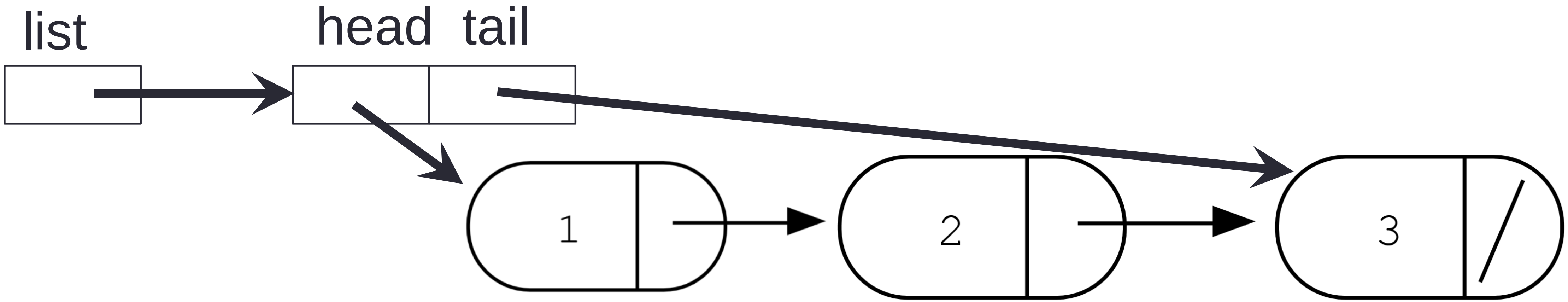
Under the hood of recursive calls (review)



```
int sumHelper(Node *head){
  if(head==NULL)
    return 0;
  return head->data+sum(head);
}
```

```
int sum(LinkedList *list){
     sumHelper(list->head);
}
```

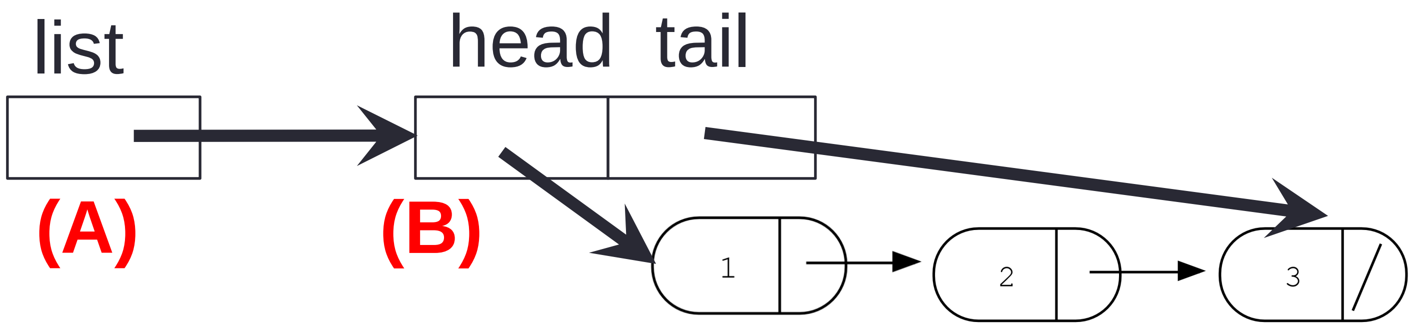
Delete node 2 in the list



Deleting the list

int freeLinkedList(LinkedList * list){…}

Which data objects are deleted by the statement: delete list;



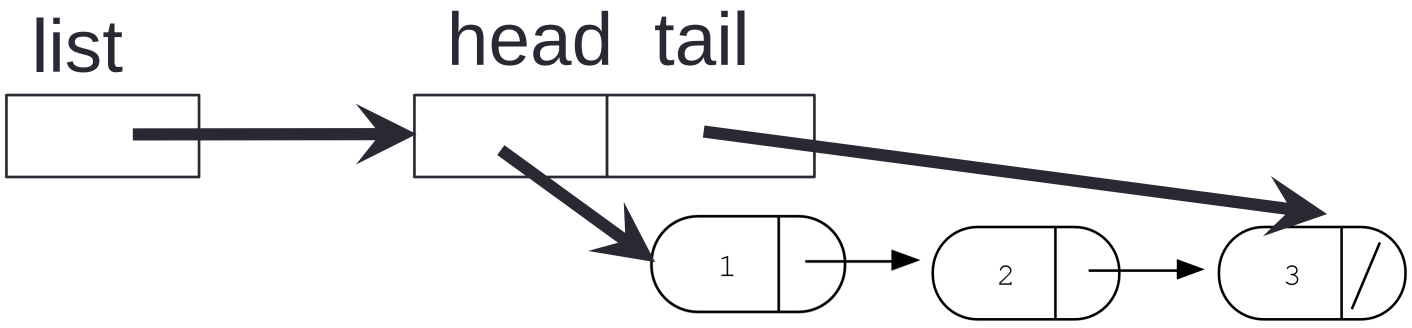
(C) All nodes of the linked list

(D) B and C
(E) All of the above

Does this result in a memory leak?

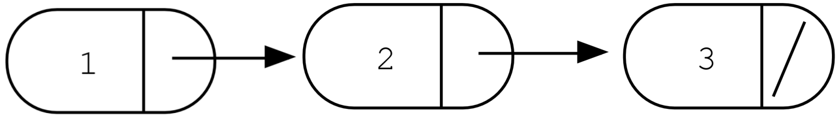
Delete the list

```
int freeLinkedList(LinkedList * list);
```



Recursion on lists: delete a value recursively

```
void deleteNodeRecursive(LinkedList *list, int value)
```



```
Node* deleteNodeRecursiveHelper(Node *head, int value)
```

Recall the steps towards a recursive solution

Next

- Final review