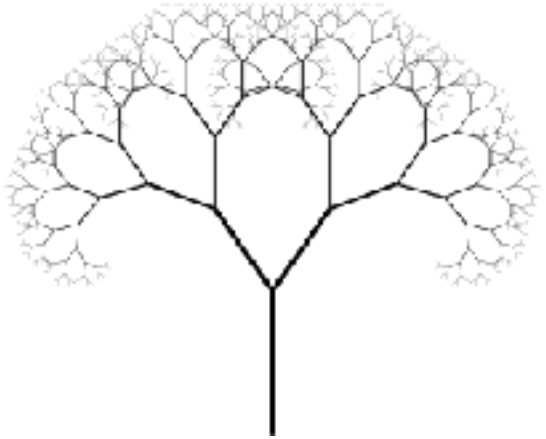
RECURSION WRAP UP

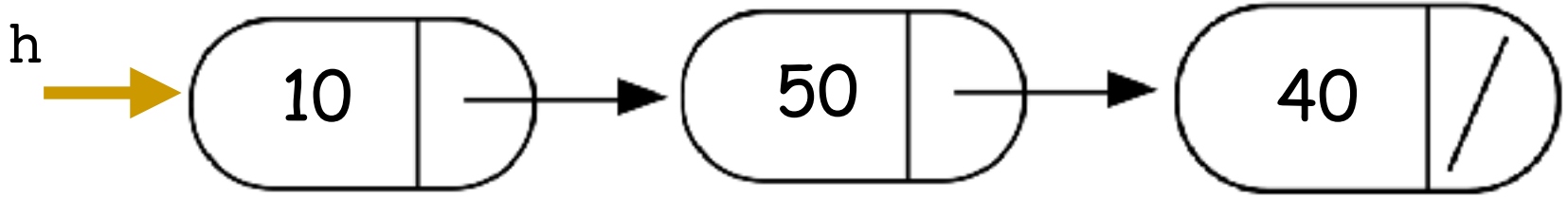
Problem Solving with Computers-I



```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```

head

sumRest



Imagine each instance of sumList to be a doll!

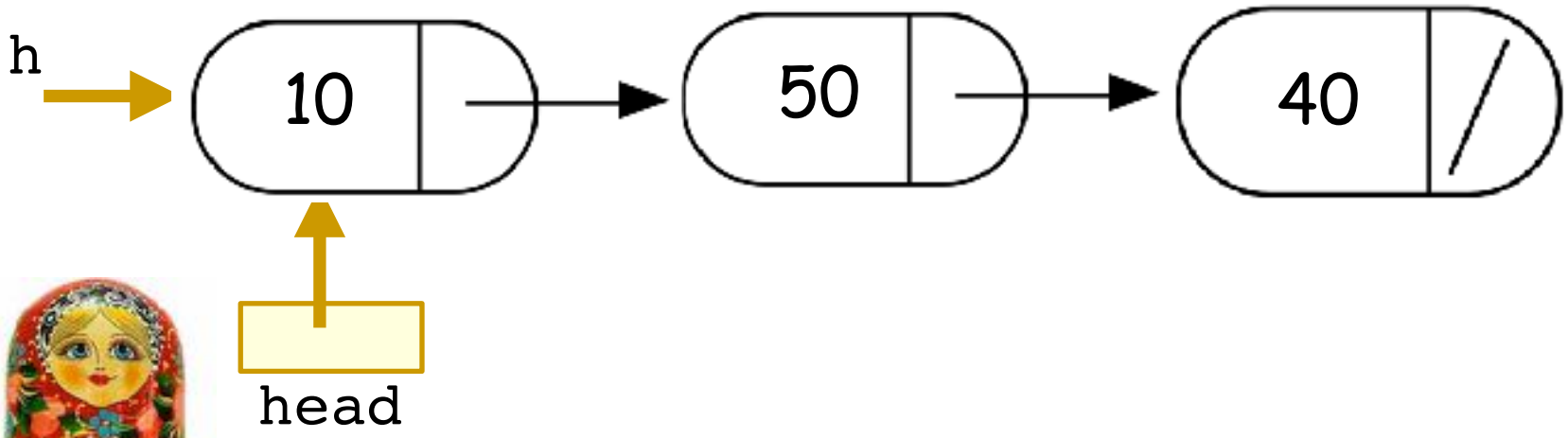
Calling sumList is like creating a new doll.

First call to sumList!

```
double s=sumList(h);
```

```
double sumList(Node* head){
    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```

The turtle tells us which line of code is going to be execute

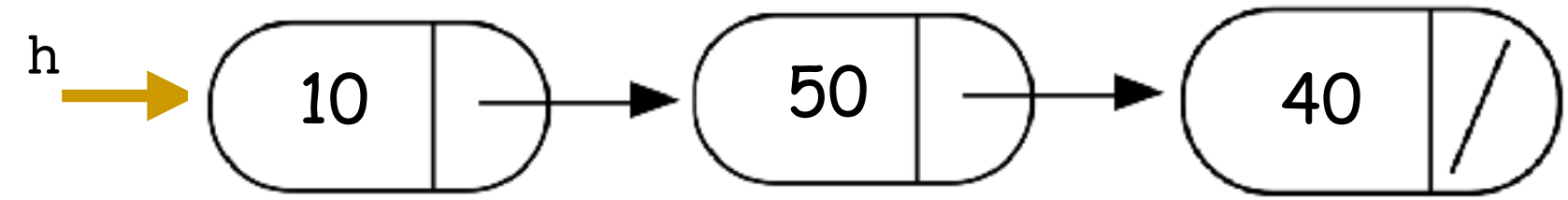


sumList(1)

First call to sumList!

```
double s=sumList(h);
```

```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```



head

?

sumRest

sumList(1)

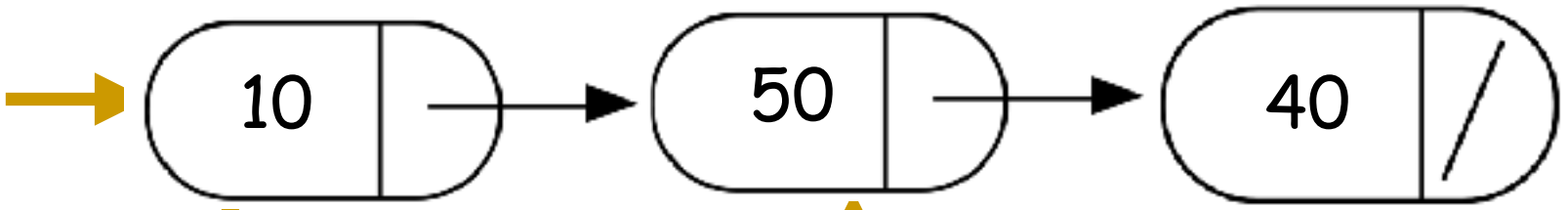
Second call to sumList!

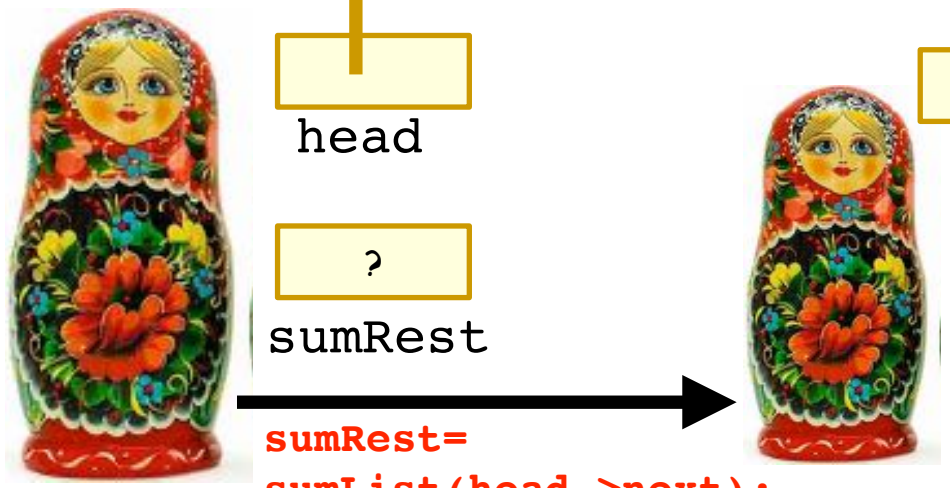
First call to sumList!

```
double s=sumList(h);
```

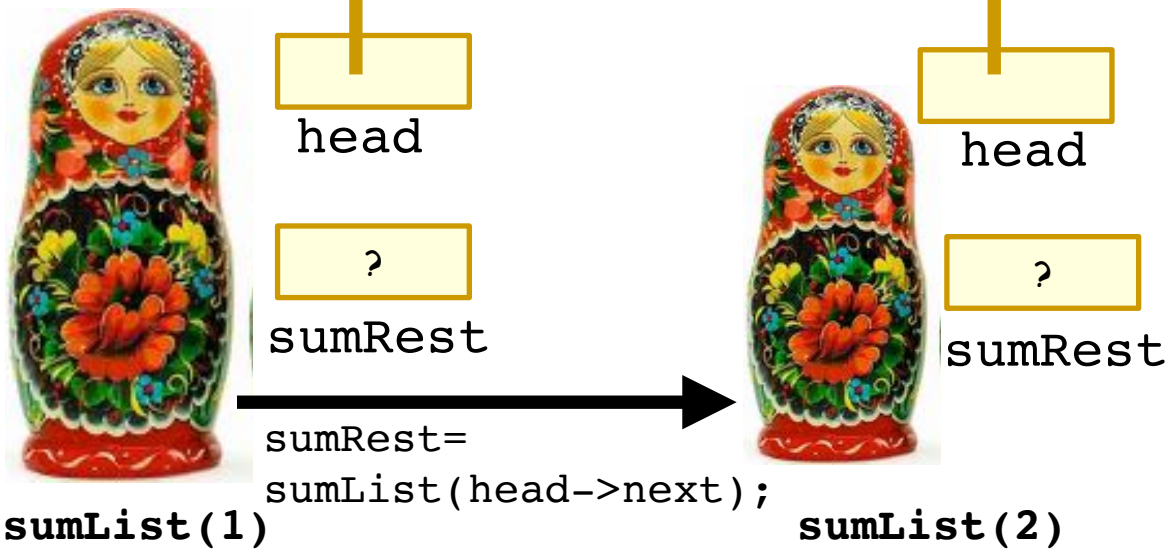
```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```

Turtle is going to execute the first line of sumList(2)



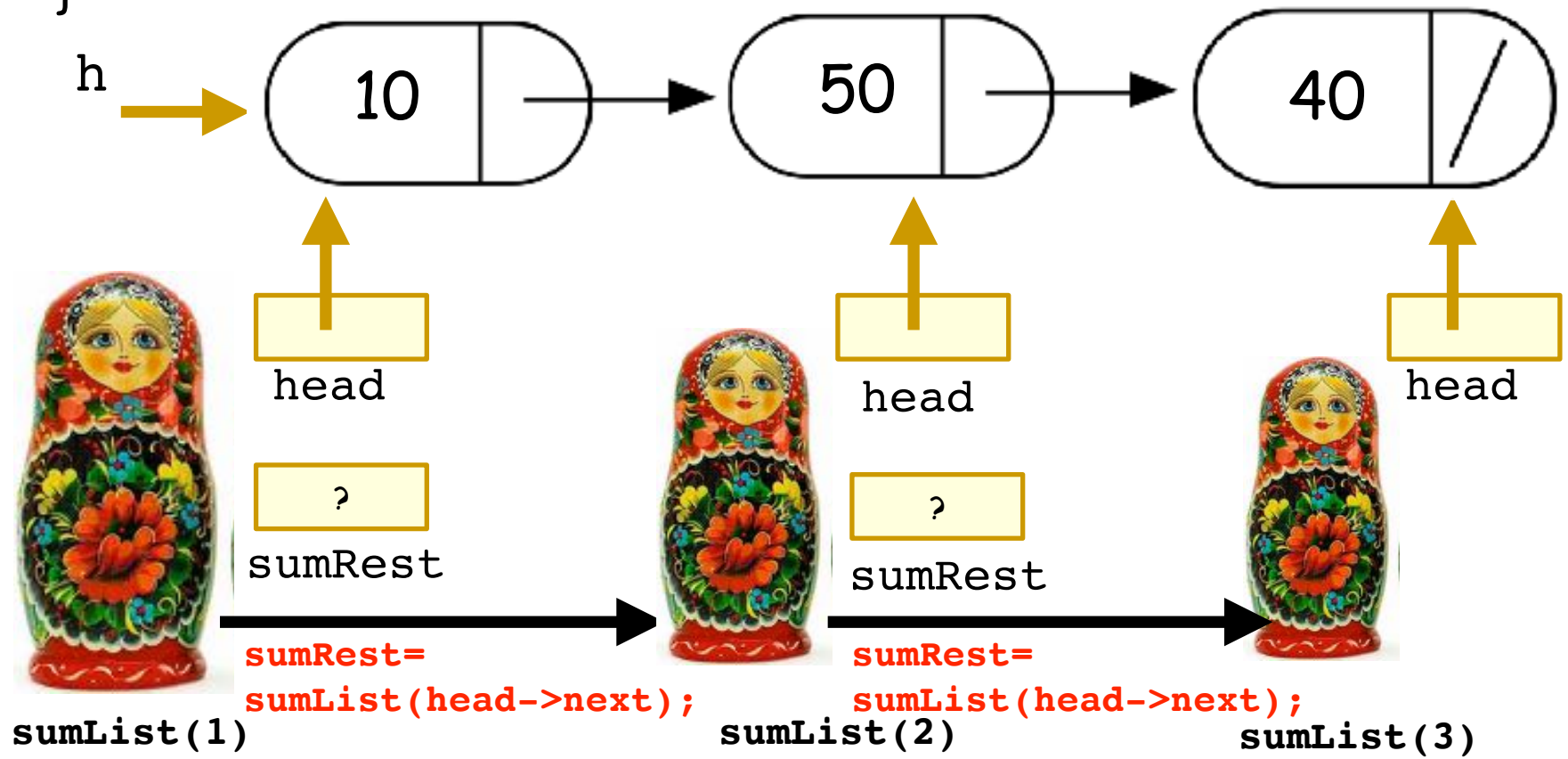


```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```

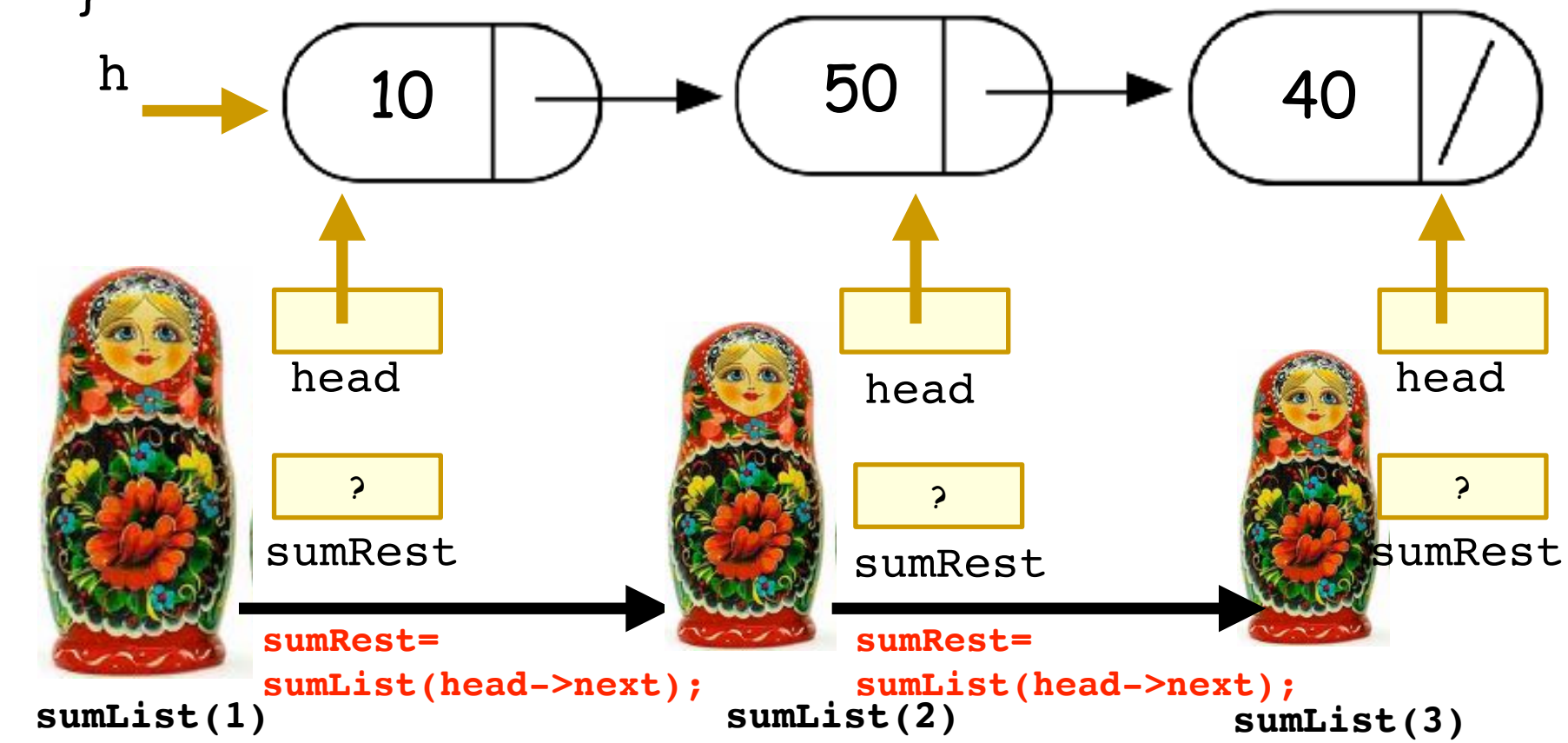


Third call to sumList

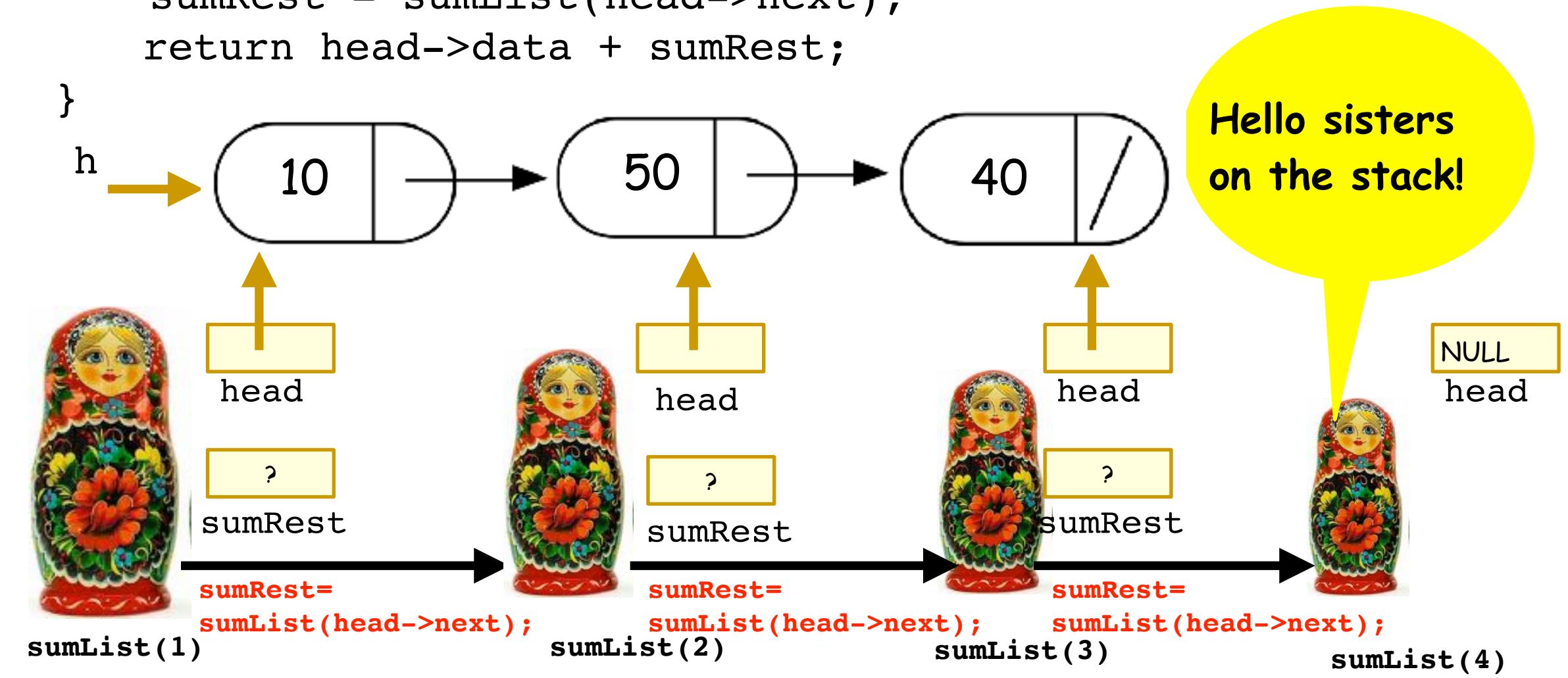
```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```



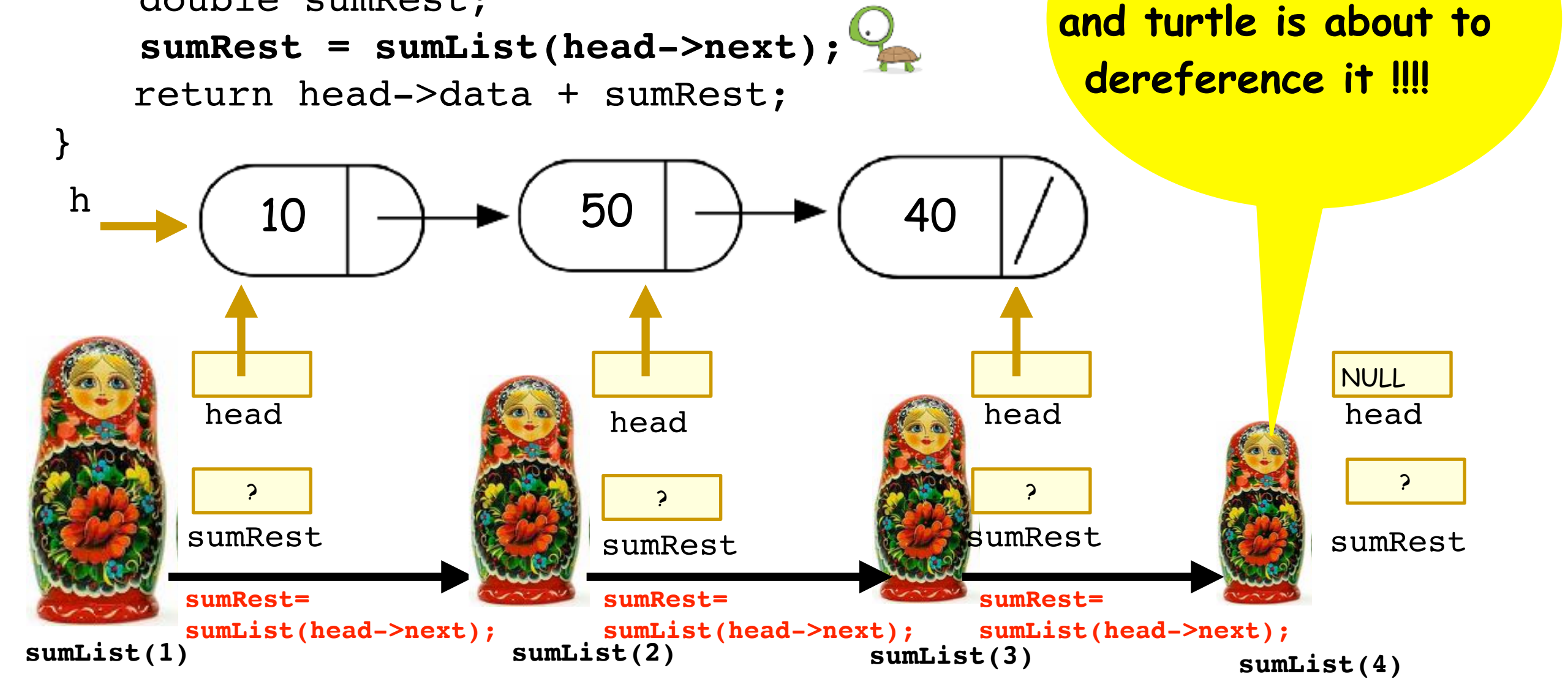
```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```



```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```

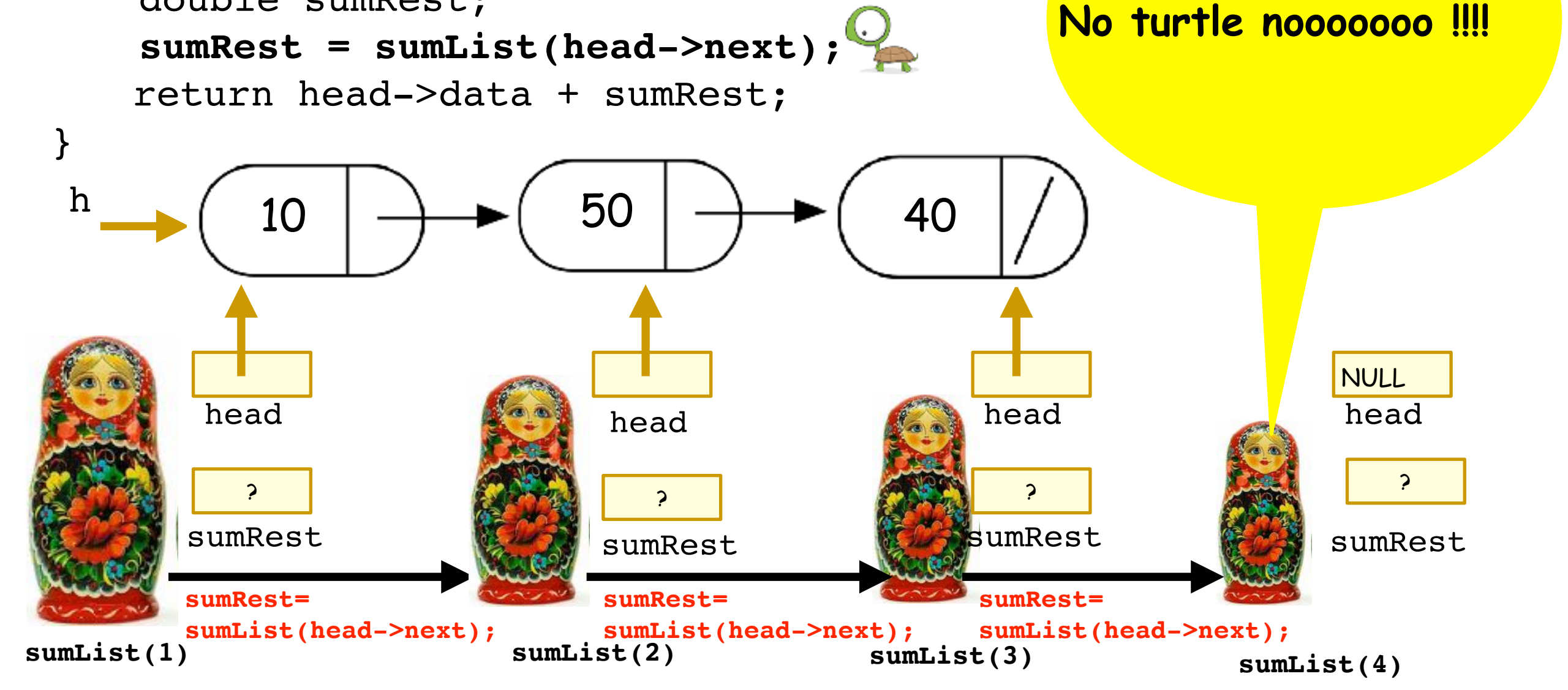


```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```

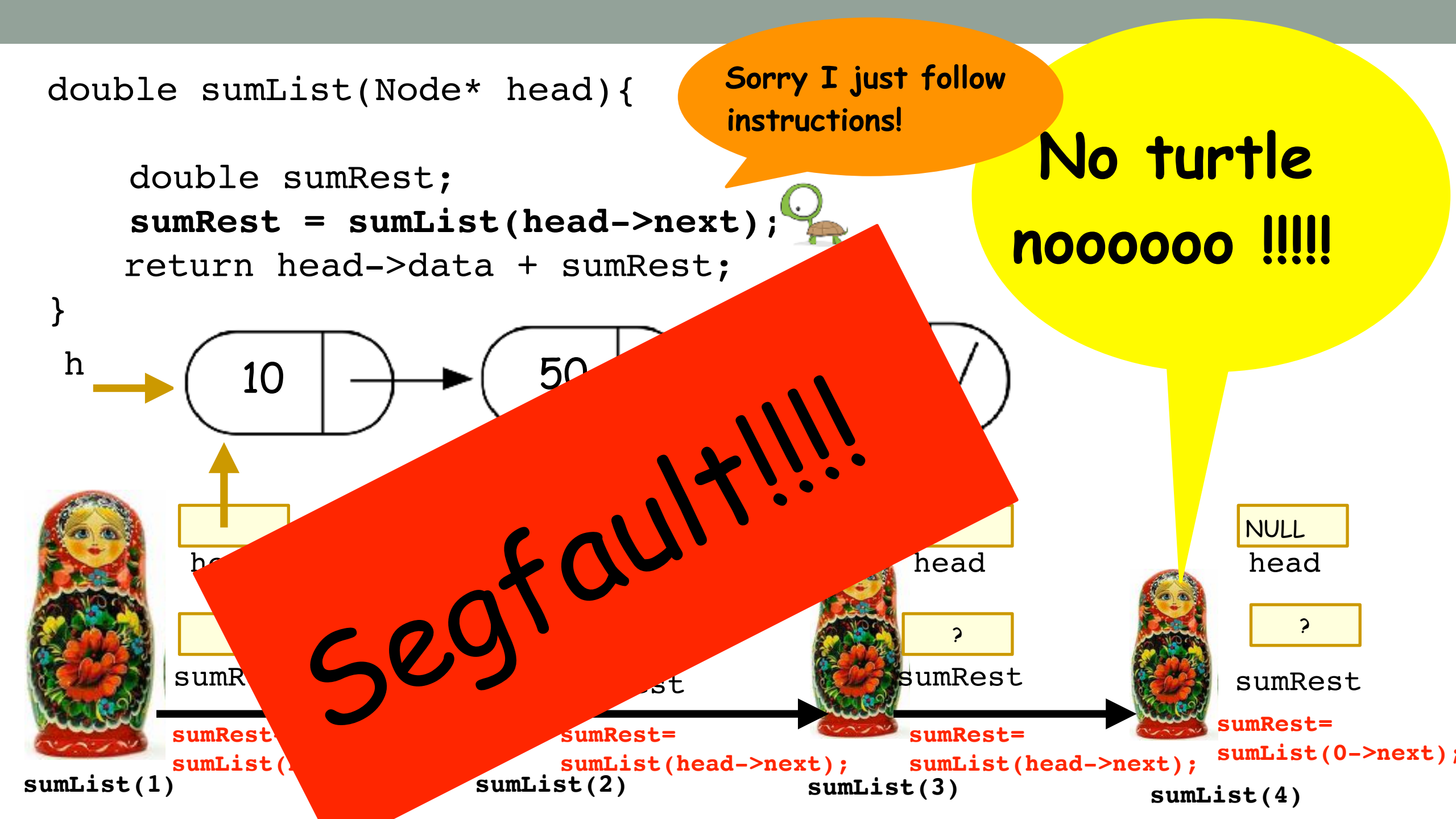


Oops my head is null and turtle is about to dereference it !!!!

```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```



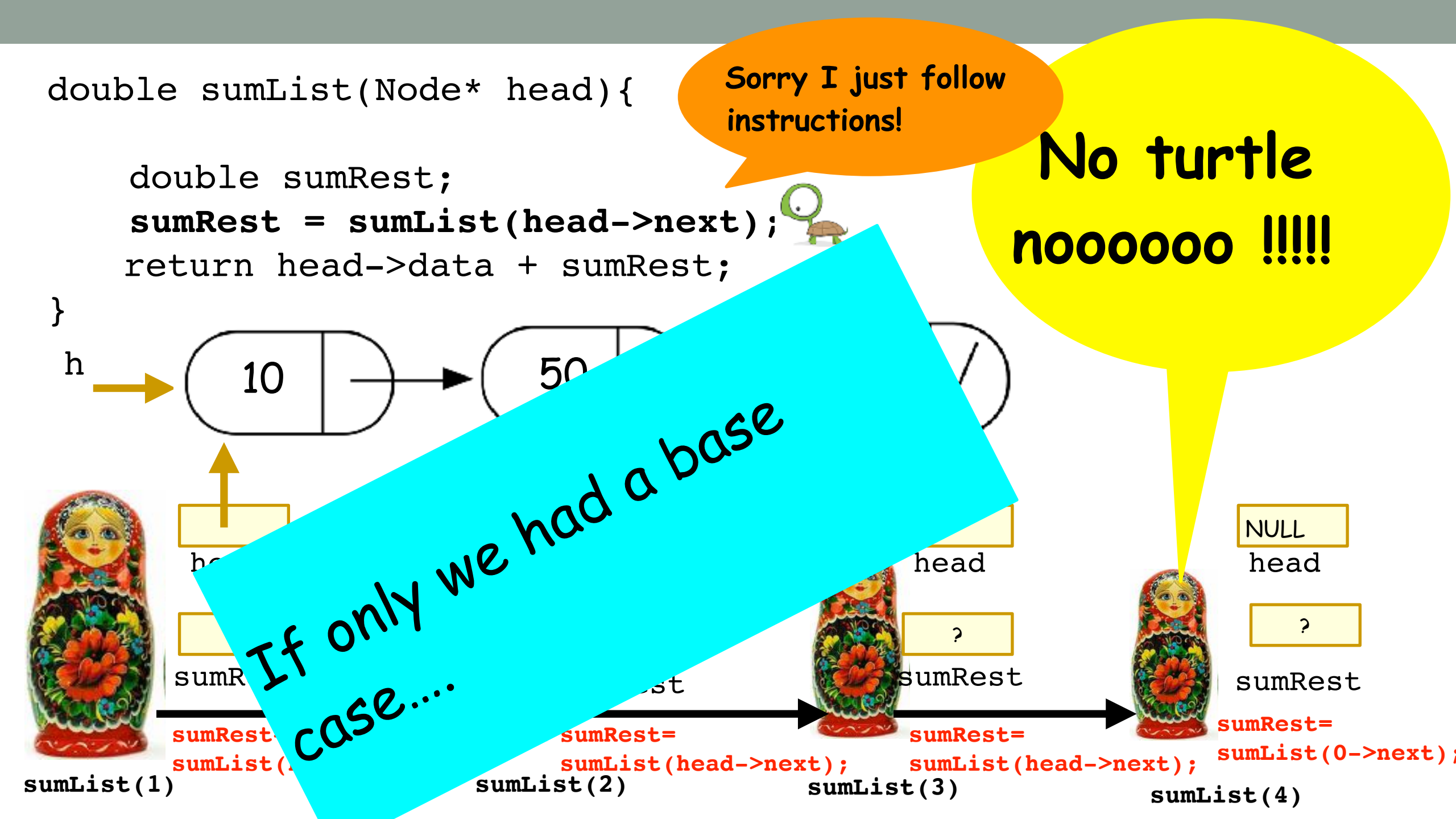
```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```



```
double sumList(Node* head){

    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```

Sorry I just follow instructions!

No turtle noooooo !!!!!

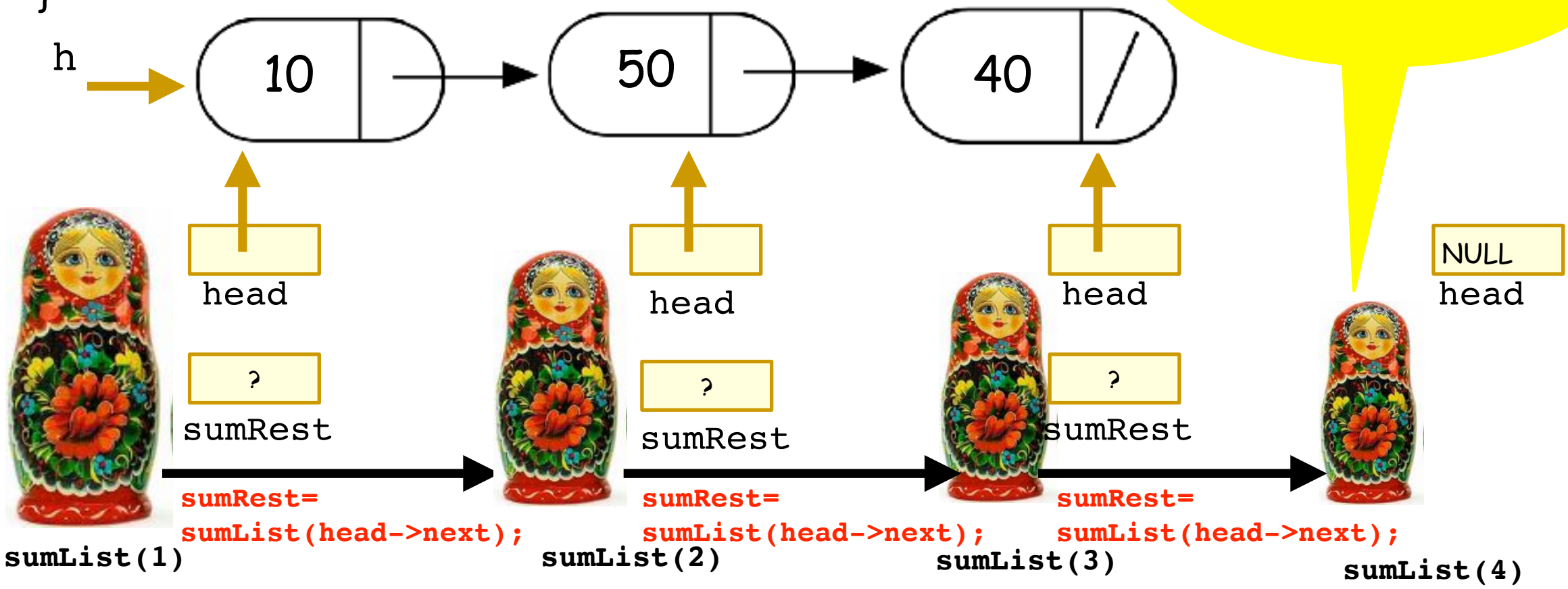


If only we had a base case....

```
double sumList(Node* head){
    if(head==0) return 0;
     double sumRest;
     sumRest = sumList(head->next);
     return head->data + sumRest;
}
```

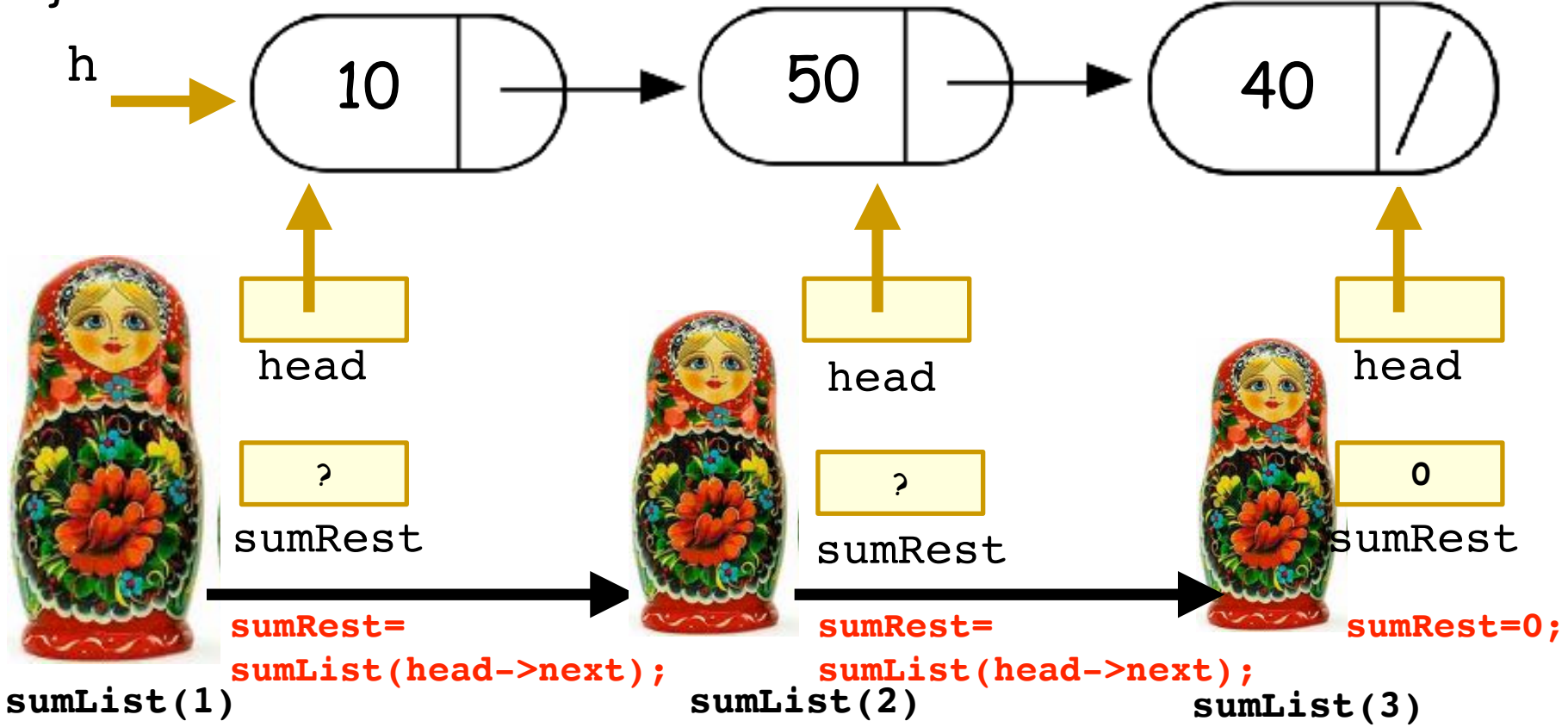
I am really well behaved around base cases :)

return 0;

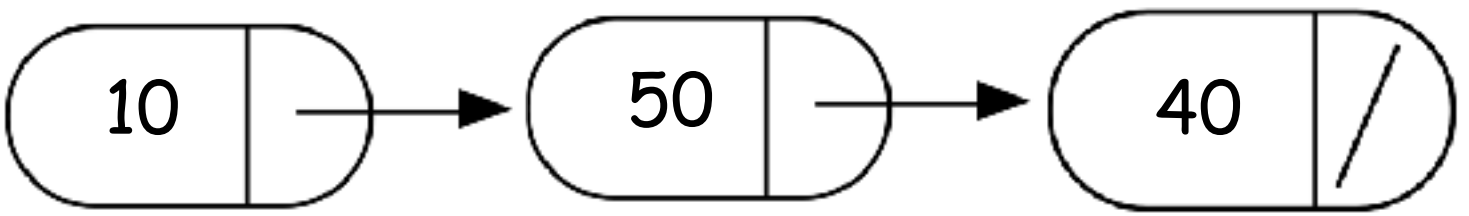


```
double sumList(Node* head){
    if(head==0) return 0;
     double sumRest;
     sumRest = sumList(head->next);
     return head->data + sumRest;
}
```

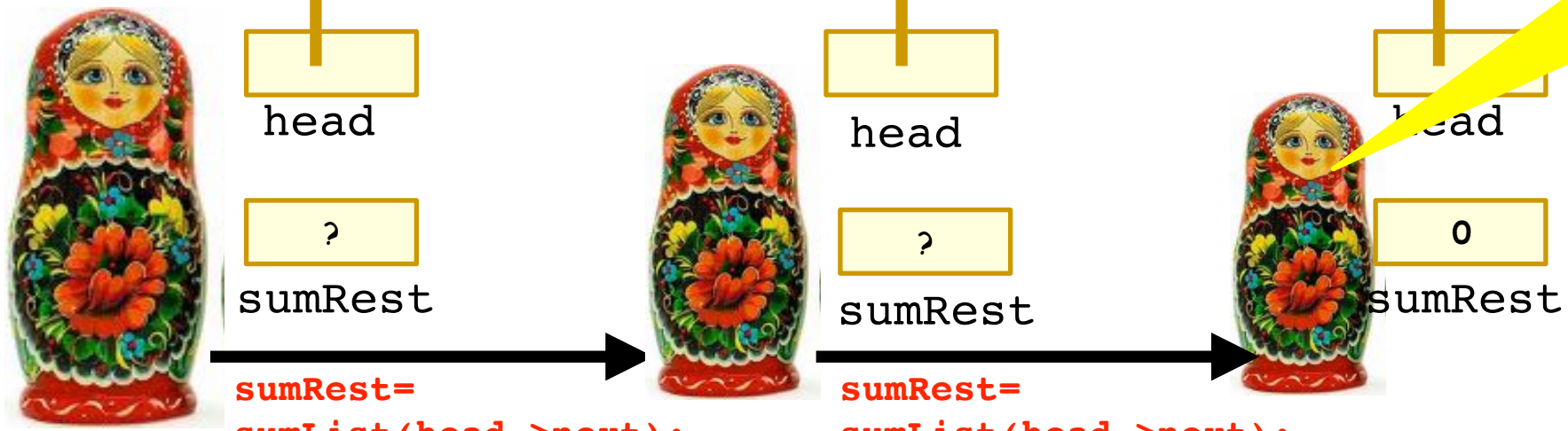
Hello again sumlist(3)! Your younger sister hit the base case.



```
double sumList(Node* head){
    if(head==0) return 0;
     double sumRest;
     sumRest = sumList(head->next);
     return head->data + sumRest;
}
```

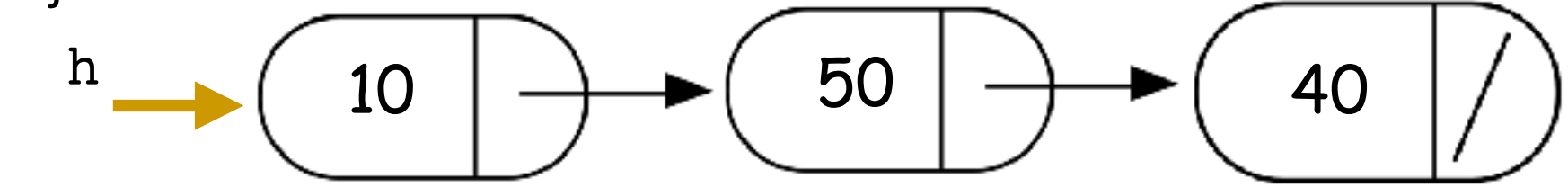


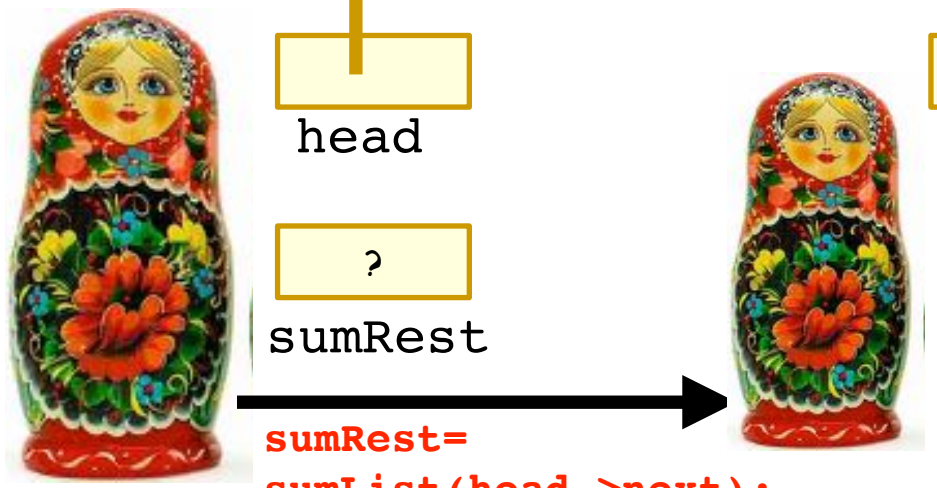
return 40+0;



```
double sumList(Node* head){
    if(head==0) return 0;
     double sumRest;
     sumRest = sumList(head->next);
     return head->data + sumRest;
}
```

Hello again Sumlist(2)! Your younger sister returned 40.

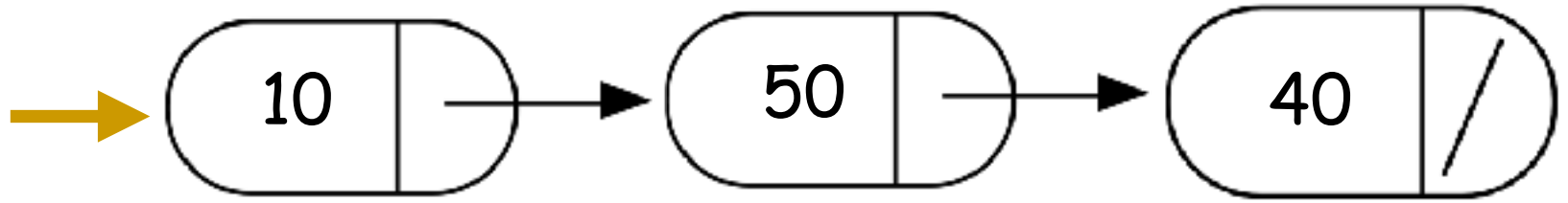





```
return head->data + sumRest;
```

```
double sumList(Node* head){
    if(head==0) return 0;
     double sumRest;
     sumRest = sumList(head->next);
     return head->data + sumRest;
}
```

Any last words, sumList(2)?



head

?

sumRest

sumRest=
sumList(head->next);

sumList(1)

head

40

sumRest

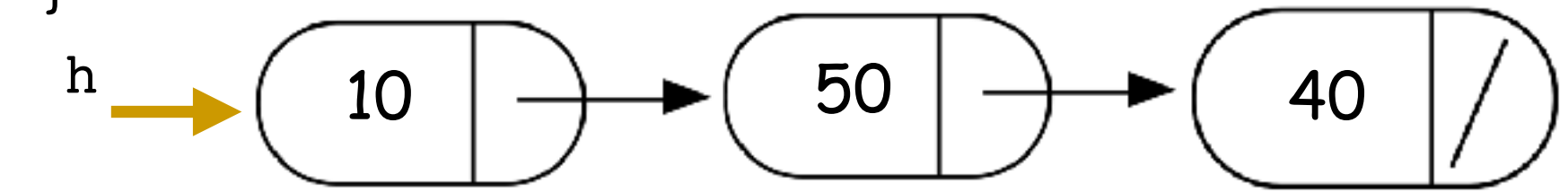
sumList(2)

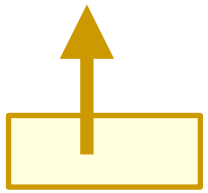
return 50+40;

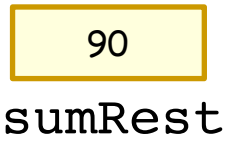
return head->data + sumRest;

```
double sumList(Node* head){
    if(head==0) return 0;
    double sumRest;
    sumRest = sumList(head->next);
    return head->data + sumRest;
}
```

Hello again sumList(1)!
Your sisters are no longer on the stack, here is your 90, store it safely!





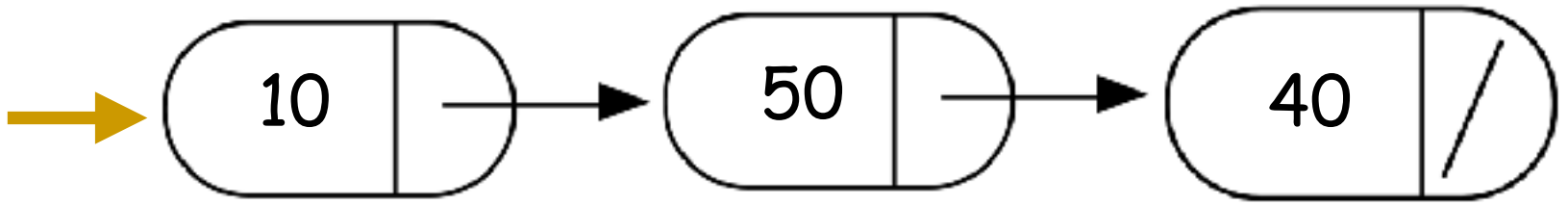


return head->data + sumRest;

sumList(1)

```
double sumList(Node* head){
    if(head==0) return 0;
     double sumRest;
     sumRest = sumList(head->next);
     return head->data + sumRest;
}
```

How did I get into this line of work, so many goodbyes make me want to cry.



head

90

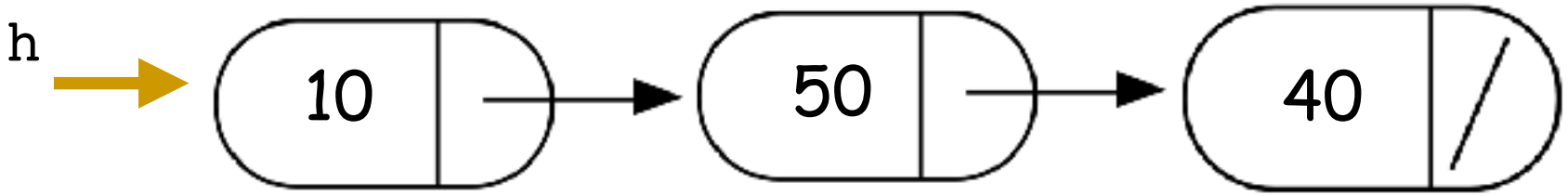
sumRest

return 10+90;

B'bye and thanks for all the computation!

sumList(1)

```
double sumList(Node* head){
    if(head==0) return 0;
     double sumRest;
     sumRest = sumList(head->next);
     return head->data + sumRest;
}
```



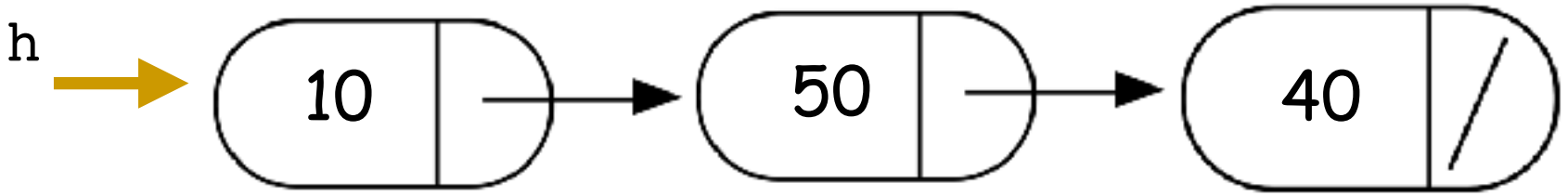
You have
no idea how many calls I had
to make to sum this list!

```
double s=sumList(h);
```

```
double sumList(Node* head){
    if(head==0) return 0;
     double sumRest;
     sumRest = sumList(head);
     return head->data + sumRest;
}
```

What happens when we call sumList on the example linke
list?

A. Returns the correct sum (100)
B. Program crashes with a segmentation fault
C. Program runs for a while, then crashes
D. None of the above



double s=sumList(h);