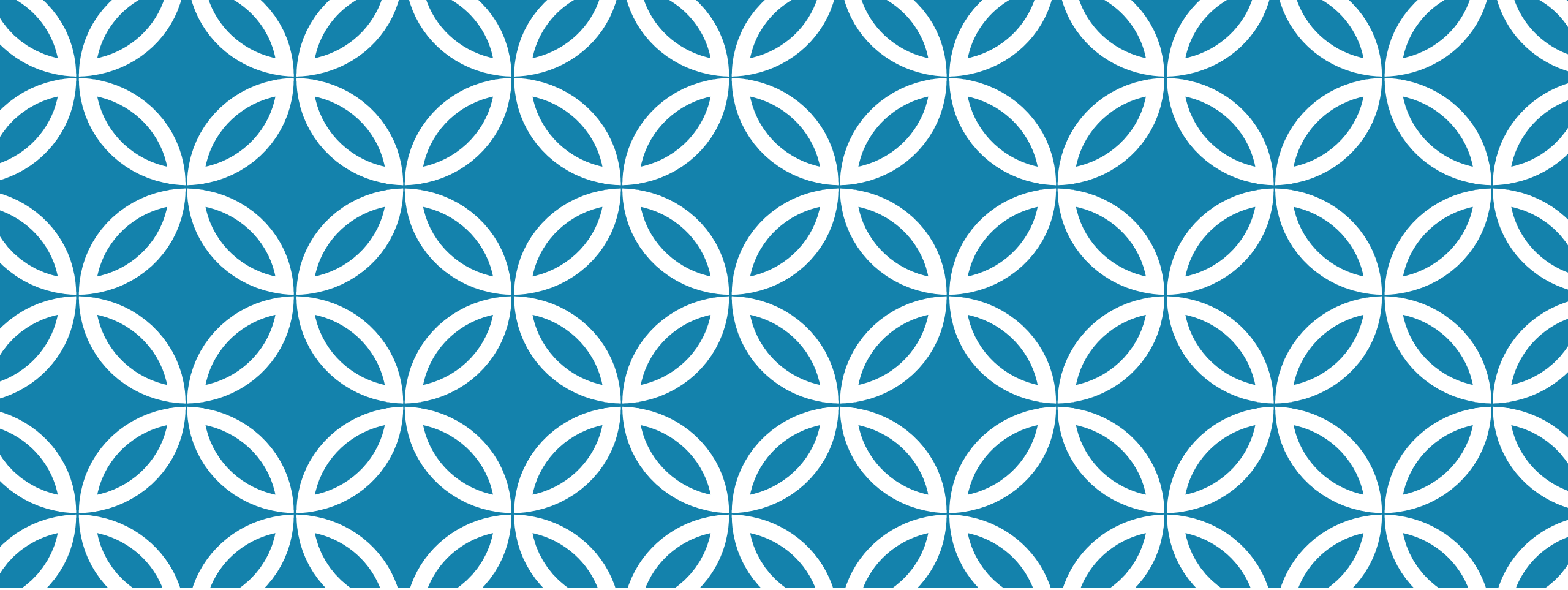
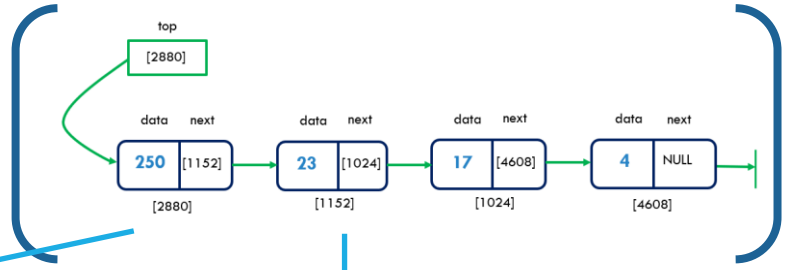


COMP 2611, Data Structures

LECTURE 4: RECURSION WITH LINKED LISTS

RECURSION TREE FOR PRINTLISTREC

printListRec



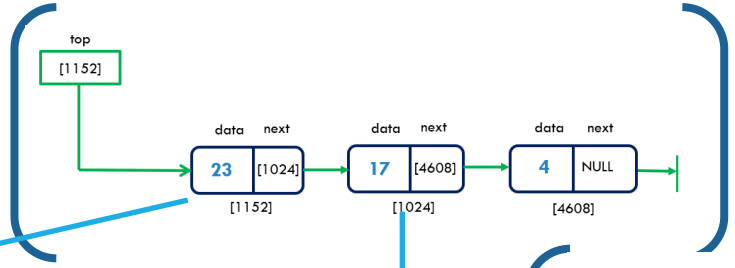
250
23
17
4

cout << 250;

printListRec

A

return

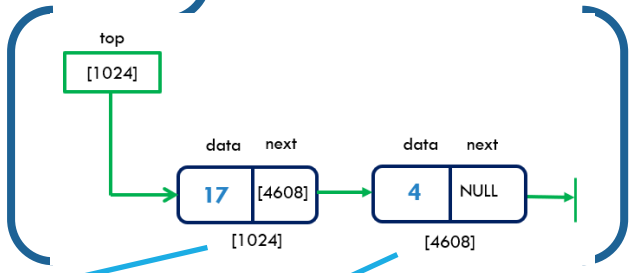


cout << 23;

printListRec

B

return

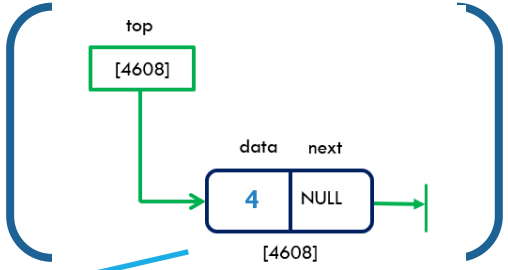


cout << 17;

printListRec

C

return



cout << 4;

printListRec (NULL)

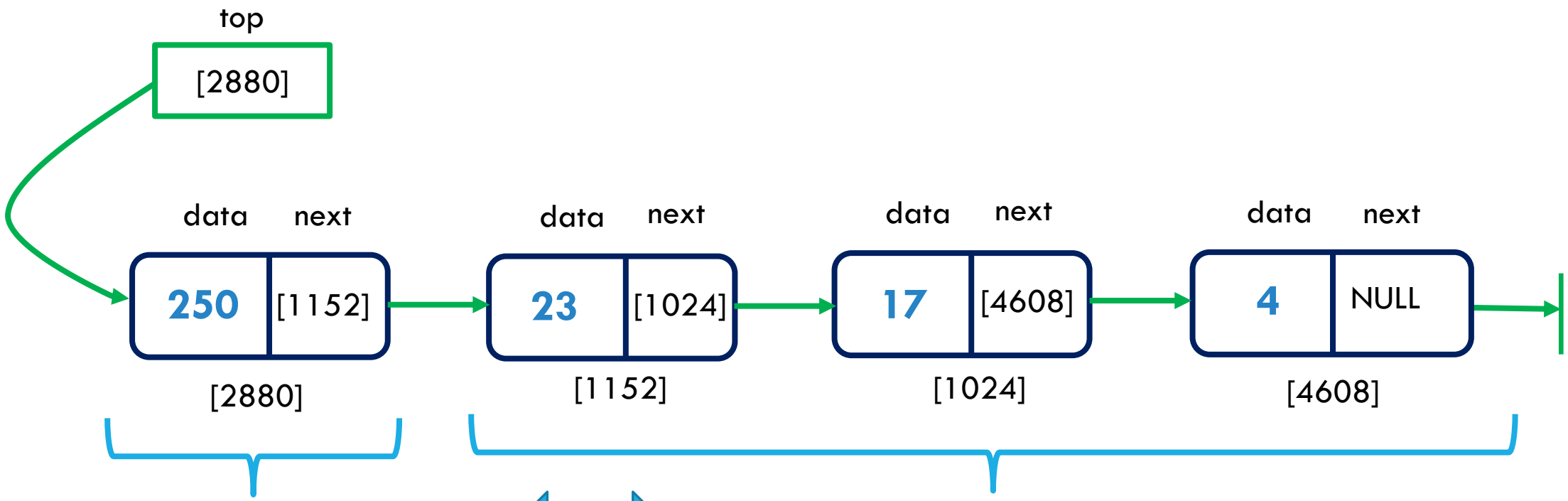
D

return

RECURSIVE "SIZE" FUNCTION

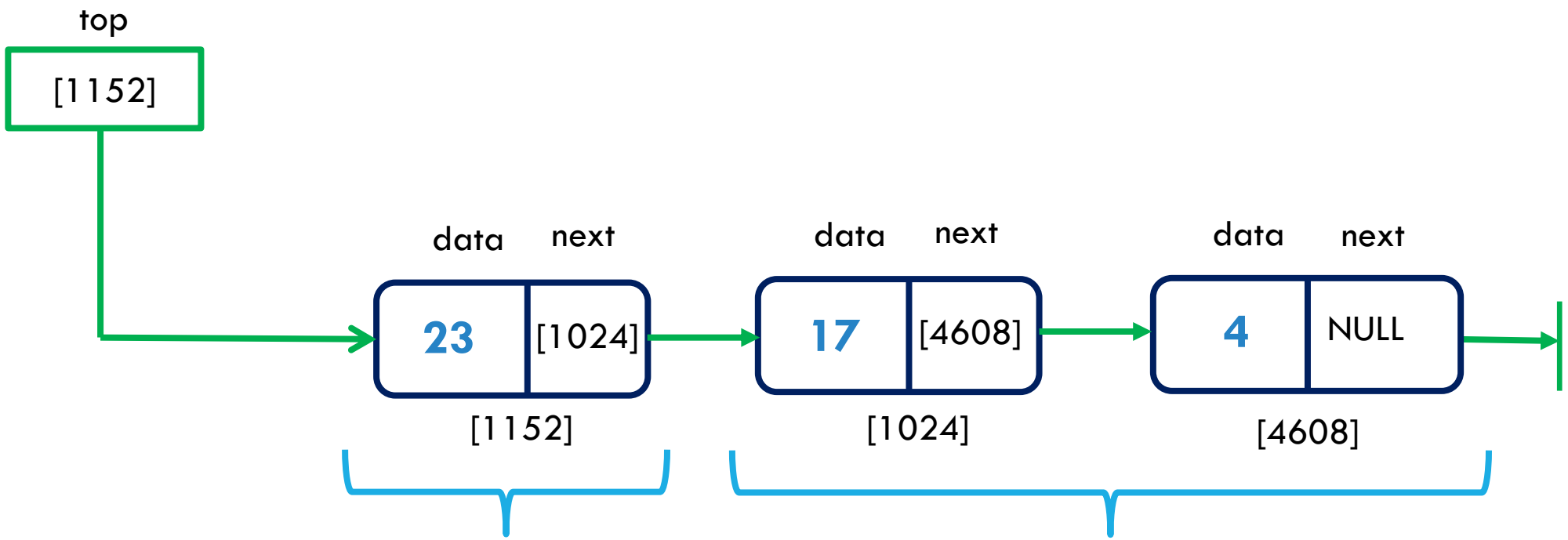
```
1 + sizeRec (top->next);
```

```
int sizeRec (Node * top);
    // finds the number of elements in a linked list
```



```
1 + (1 + sizeRec (top->next));
```

RECURSIVE CALL (#1)



count first element:

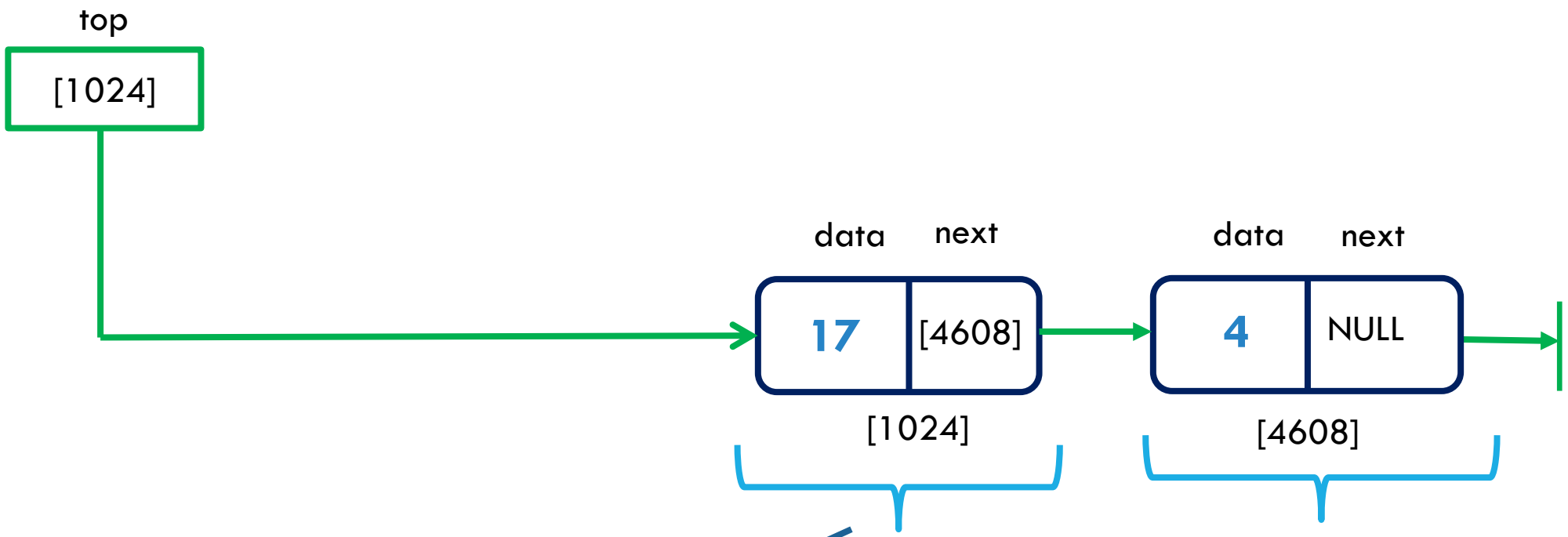
1

count remaining elements:

`sizeRec(top->next);`

```
1 + (1 + (1 + sizeRec (top->next)));
```

RECURSIVE CALL (#2)



count first element:

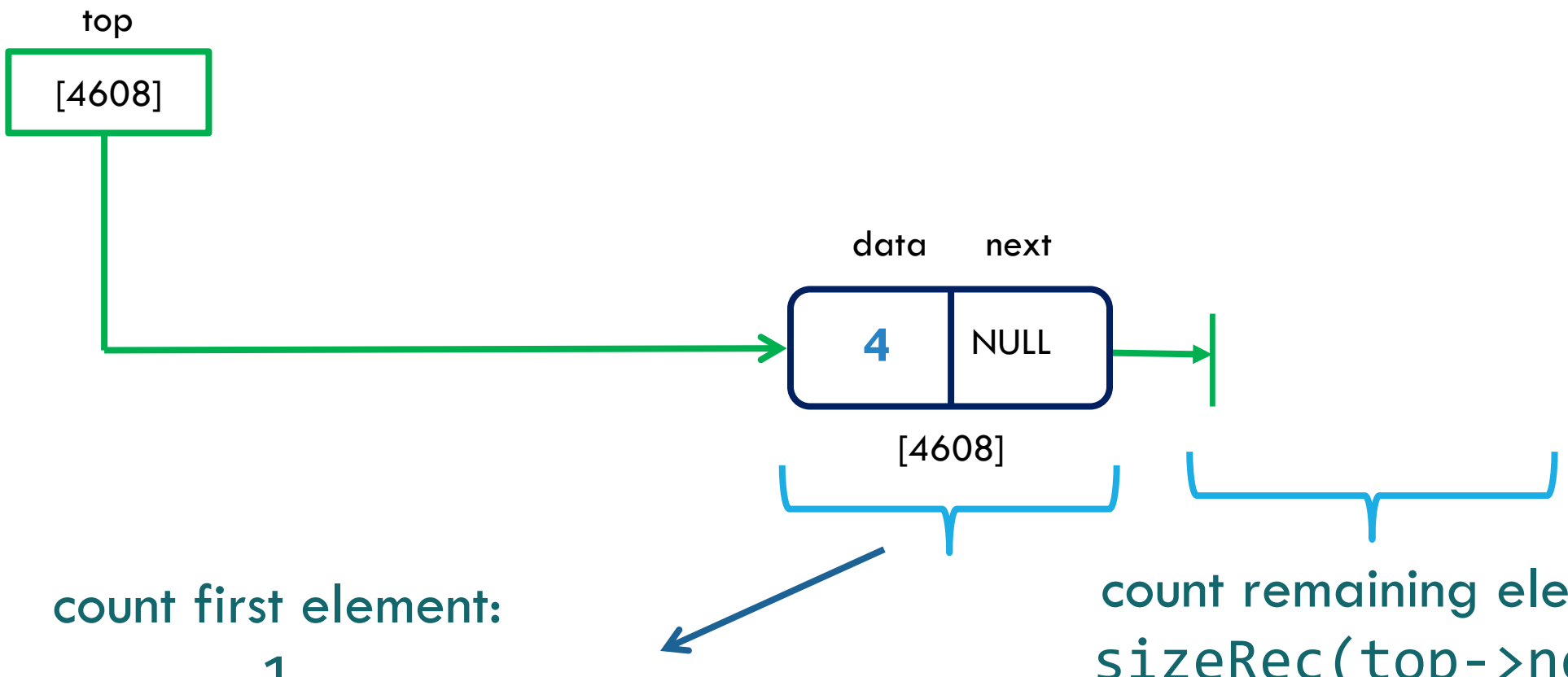
1

count remaining elements:

`sizeRec(top->next);`

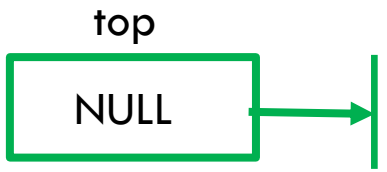
```
1 + (1 + (1 + (1 + sizeRec (top->next))));
```

RECURSIVE CALL (#3)



RECURSIVE CALL (#4)

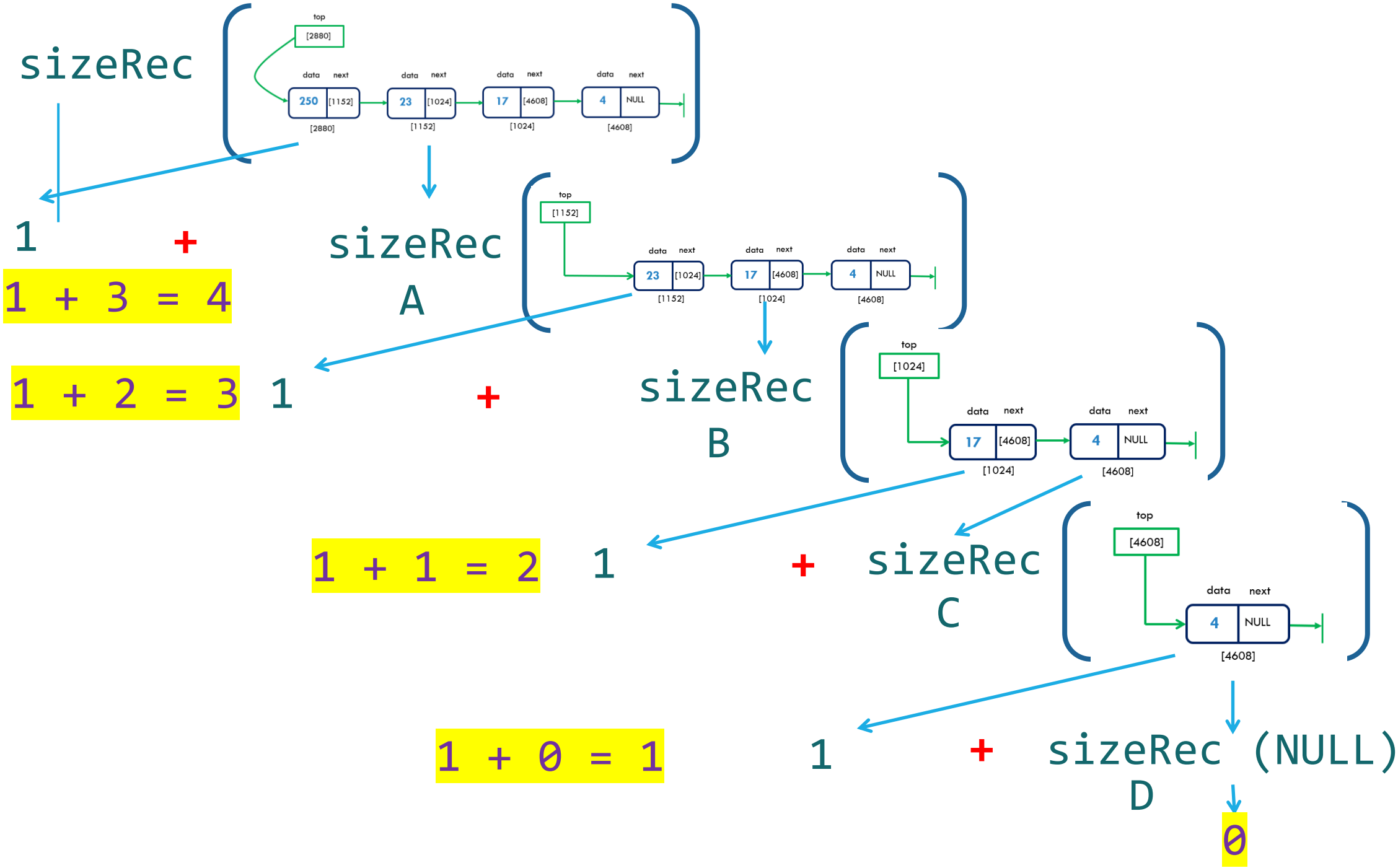
```
1 + (1 + (1 + (1 + ( 0 ))))
```



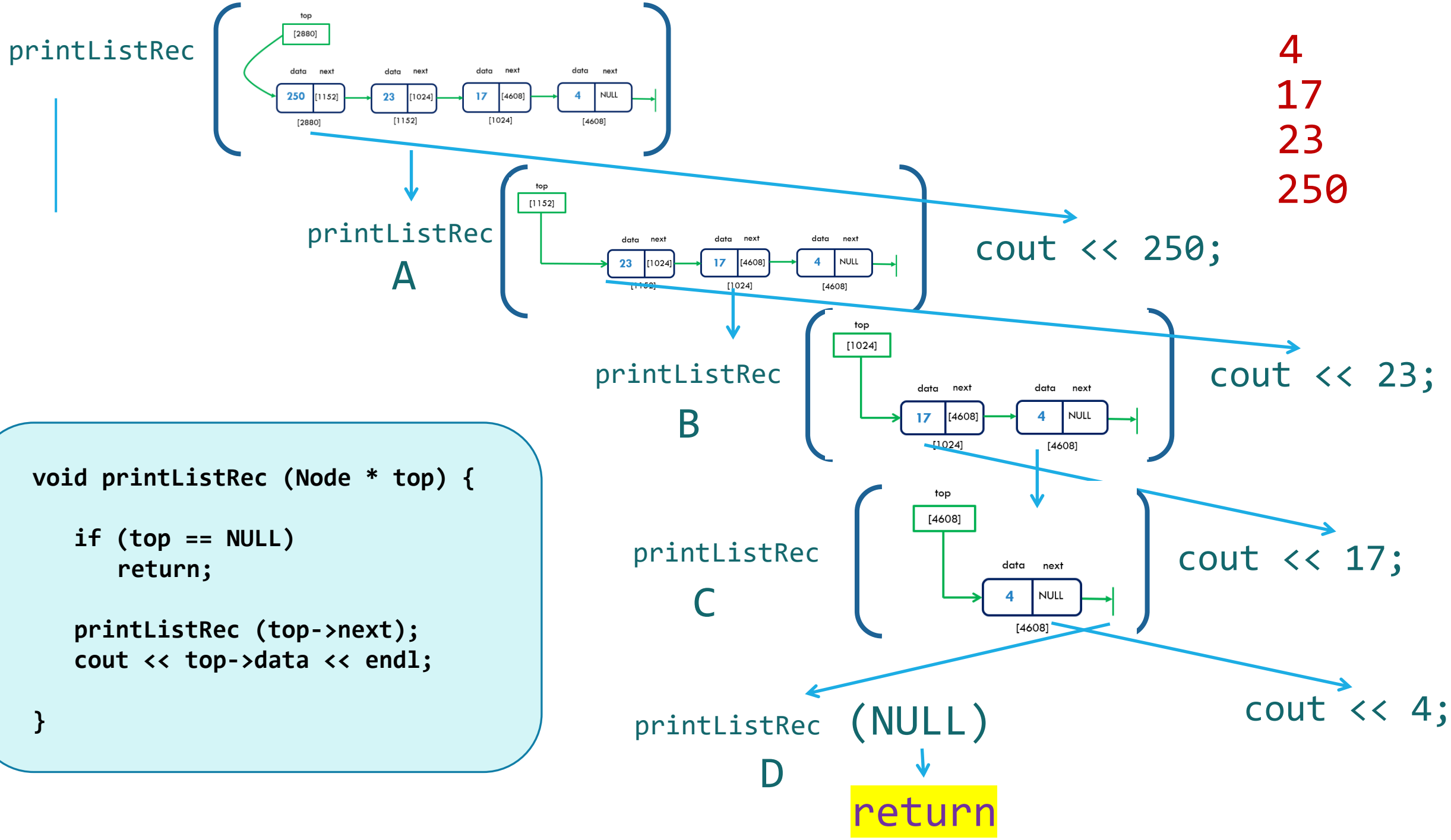
Since the list is empty,
do nothing and return 0.

```
int sizeRec (Node * top) {
    if (top == NULL)
        return 0;

    return 1 + sizeRec(top->next);

}
```

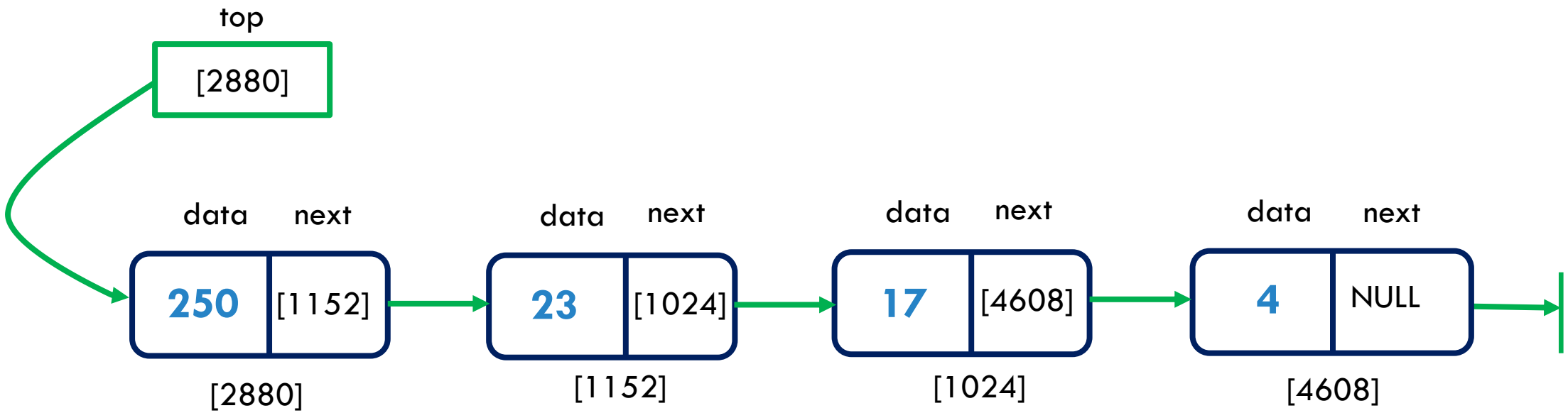
RECURSION TREE FOR SIZEREC


sizeRec
top
[2880]
data next
250 [1152]
[2880]
23 [1024]
[1152]
17 [4608]
[1024]
4 NULL
[4608]
1 +
1 + 3 = 4
sizeRec
A
top
[1152]
23 [1024]
[1152]
17 [4608]
[1024]
4 NULL
[4608]
1 + 2 = 3
1 +
sizeRec
B
top
[1024]
17 [4608]
[1024]
4 NULL
[4608]
1 + 1 = 2
1 +
sizeRec
C
top
[4608]
4 NULL
[4608]
1 + 0 = 1
1 +
sizeRec (NULL)
D
0

PRINTLISTREC (PRINT ELEMENTS IN REVERSE ORDER)


printListRec
top
[2880]
data next
250 [1152]
23 [1024]
17 [4608]
4 NULL
[2880]
[1152]
[1024]
[4608]
printListRec
A
top
[1152]
23 [1024]
17 [4608]
4 NULL
[1024]
[4608]
printListRec
B
top
[1024]
17 [4608]
4 NULL
[1024]
[4608]
printListRec
C
top
[4608]
4 NULL
[4608]
printListRec (NULL)
D
return
cout << 250;
cout << 23;
cout << 17;
cout << 4;
4
17
23
250
void printListRec (Node * top) {
if (top == NULL)
return;
printListRec (top->next);
cout << top->data << endl;
}

PRINTLISTREVERSE



How to display elements in reverse order
WITHOUT using recursion?