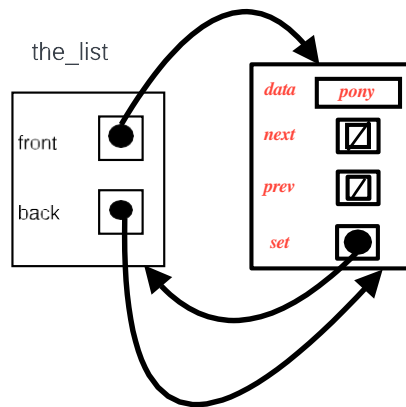What does the following function call do to this object?
What does the function return?
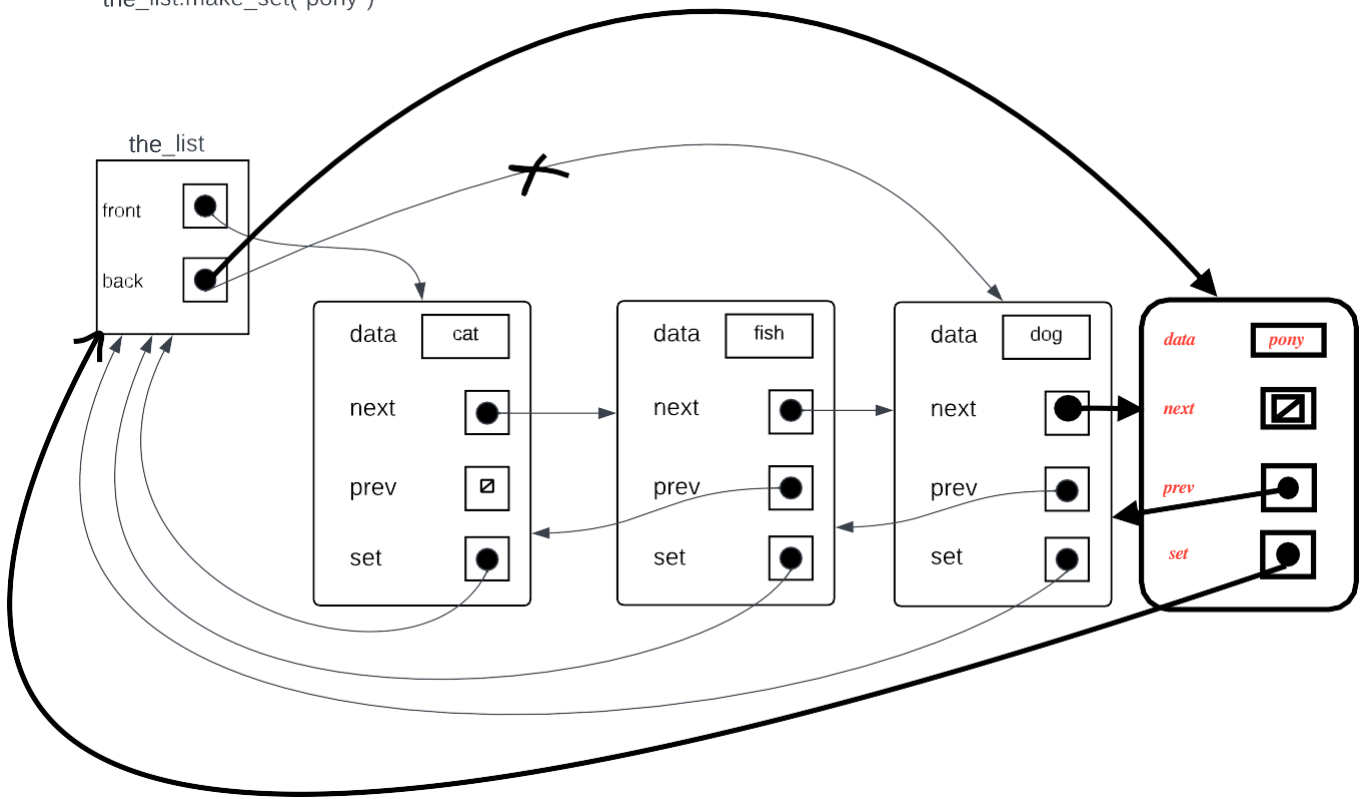
the_list.make_set("pony")

the_list

front	●
back	●

data	*pony*
next	▨
prev	▨
set	●

Make_set() is called on the "the_list" object, which is initially empty, and produces a node with the value "pony" for the "data" property. Both the 'front' and 'back' relate to newly formed nodes. There is now just one node in the the_list object; the next and previous nodes of a newly generated node will both be None. Also, the_list being originally empty, make_set() is going to give a reference to the newly generated node.

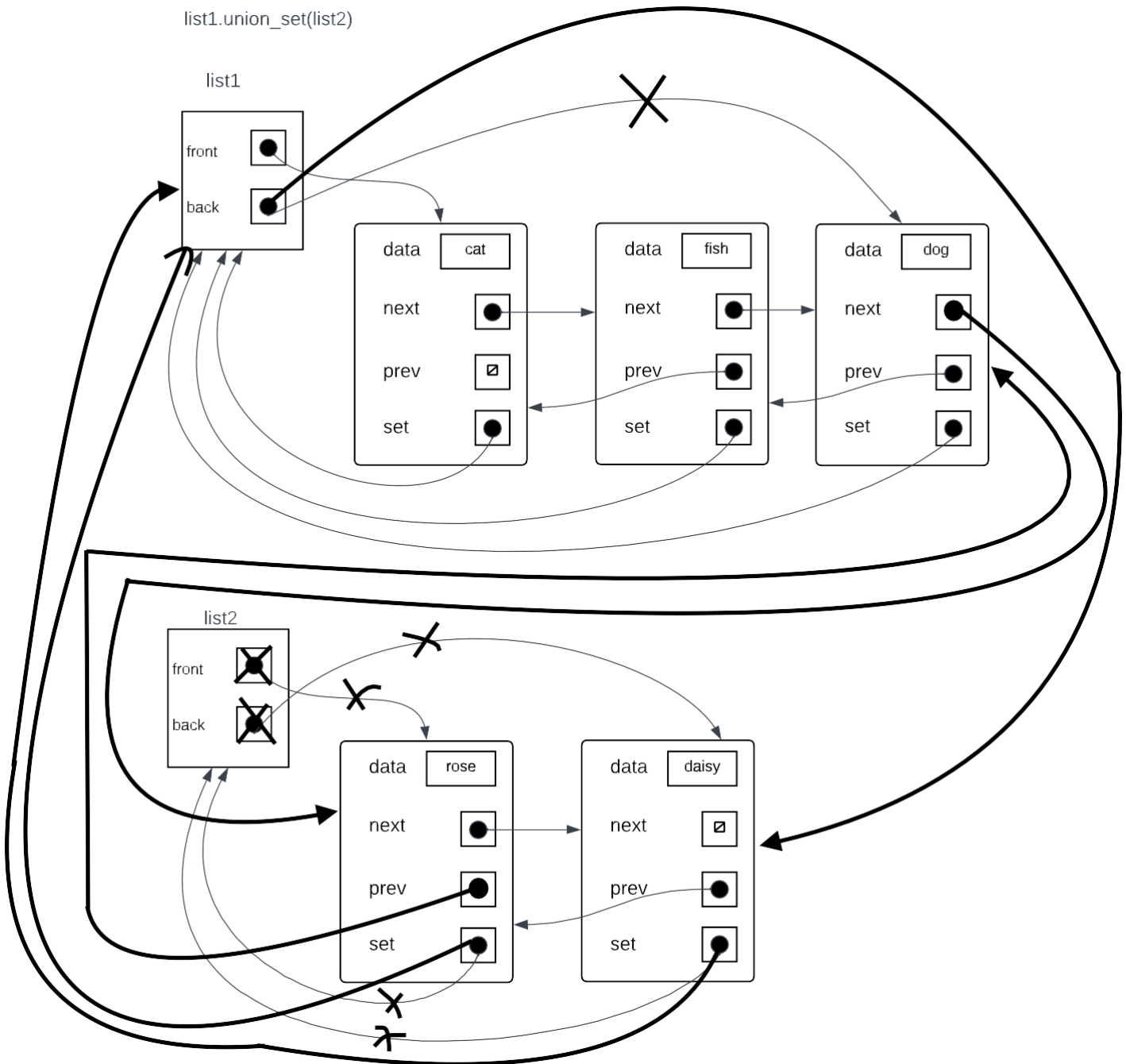What does the following function call do to this object?
What does the function return?

the_list.make_set("pony")

The make_set() function looks through the nodes' "data" properties in the "the_list" object. Whenever it discovers a match, it returns false. If no matching data is found, a new node is created with the data assign the value "pony," the next attribute set to "None," the previous node attribute pointing to the node with data "dog," and the back attribute of the_list object pointing to the newly created node. The function then returns true .

What does the following function call do to these objects?
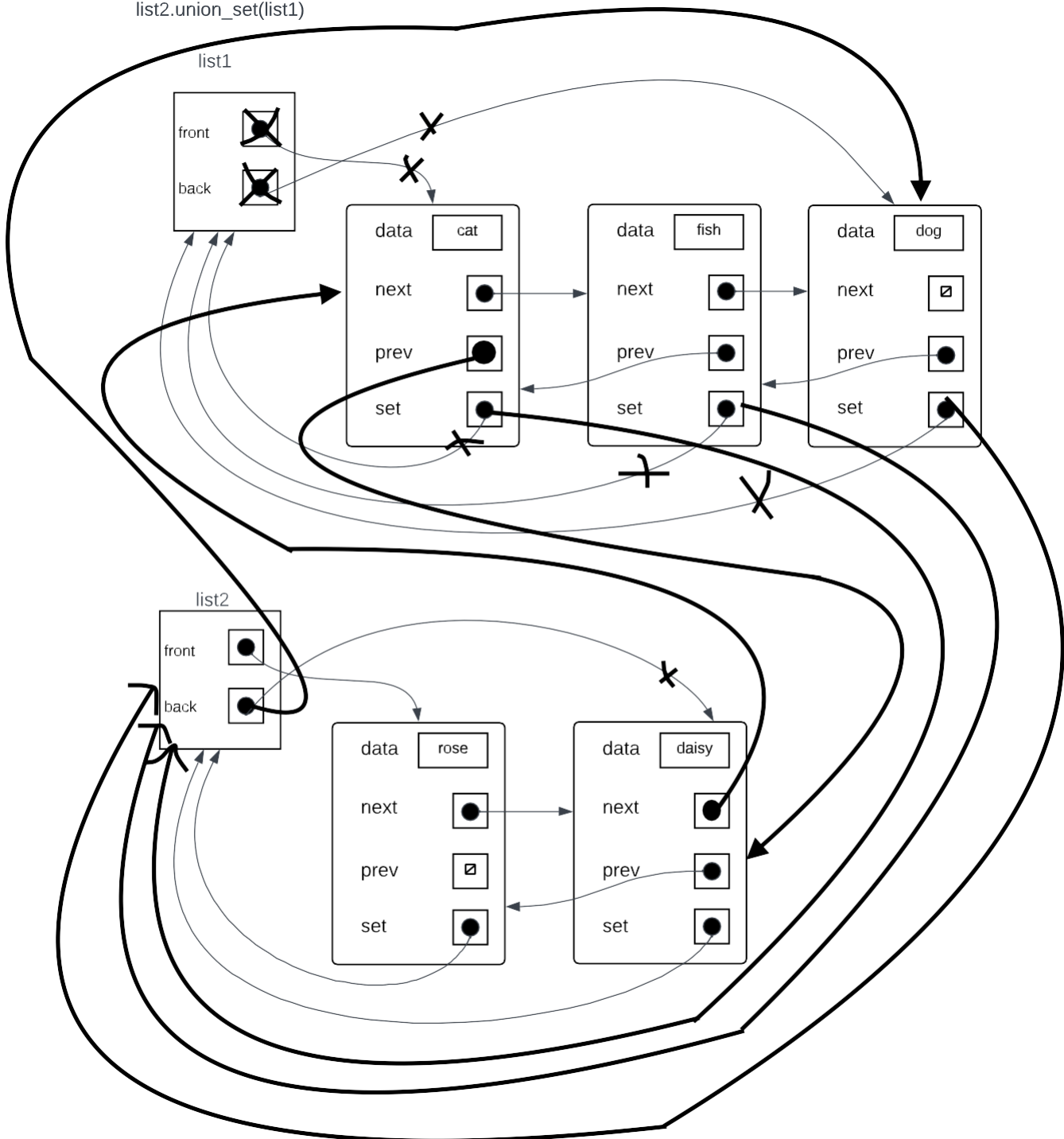What does the function return?

list1.union_set(list2)

list1

| front | ● |
| back | ● |

data | cat
next | ●
prev | ∅
set | ●

data | fish
next | ●
prev | ●
set | ●

data | dog
next | ●
prev | ●
set | ●

list2

| front | ✖ |
| back | ✖ |

data | rose
next | ●
prev | ●
set | ●

data | daisy
next | ∅
prev | ●
set | ●

Calling the *union_set()* *method will combine list2 with list1. All'set' properties of nodes in list2 will begin pointing towards list1 object after function call. The prior attribute of the front of list2 will start pointing to the rear of list1 when the next attribute of the back of list1 starts linking to the front attribute of list2.List2's back attribute will be updated from List1's back. List2's front and back characteristics will be changed to "None."*

What does the following function call do to these objects?
What does the function return?

list2.union_set(list1)

list1

| front | |
| back | |

data | cat
next | ●
prev | ●
set | ●

data | fish
next | ●
prev | ●
set | ●

data | dog
next | ☑
prev | ●
set | ●

list2

| front | ● |
| back | ● |

data | rose
next | ●
prev | ☑
set | ●

data | daisy
next | ●
prev | ●
set | ●

List1 and List2 will be combined using *union_set().* *All'set' properties of every node in list1 will start pointing towards list2 object after the function call. List2 object's back property will initially link to List1's back. a further characteristic of the back 'prev' property on the front of list1 will start pointing towards the rear of list2, and the front of list2 will start pointing towards the front of list1.* *Union_set()* *returns the number of nodes passed on, which is equivalent to list1's width.'None' will be assigned as the value for the list1 object's front and back properties.*

What does the following function call do to these objects?
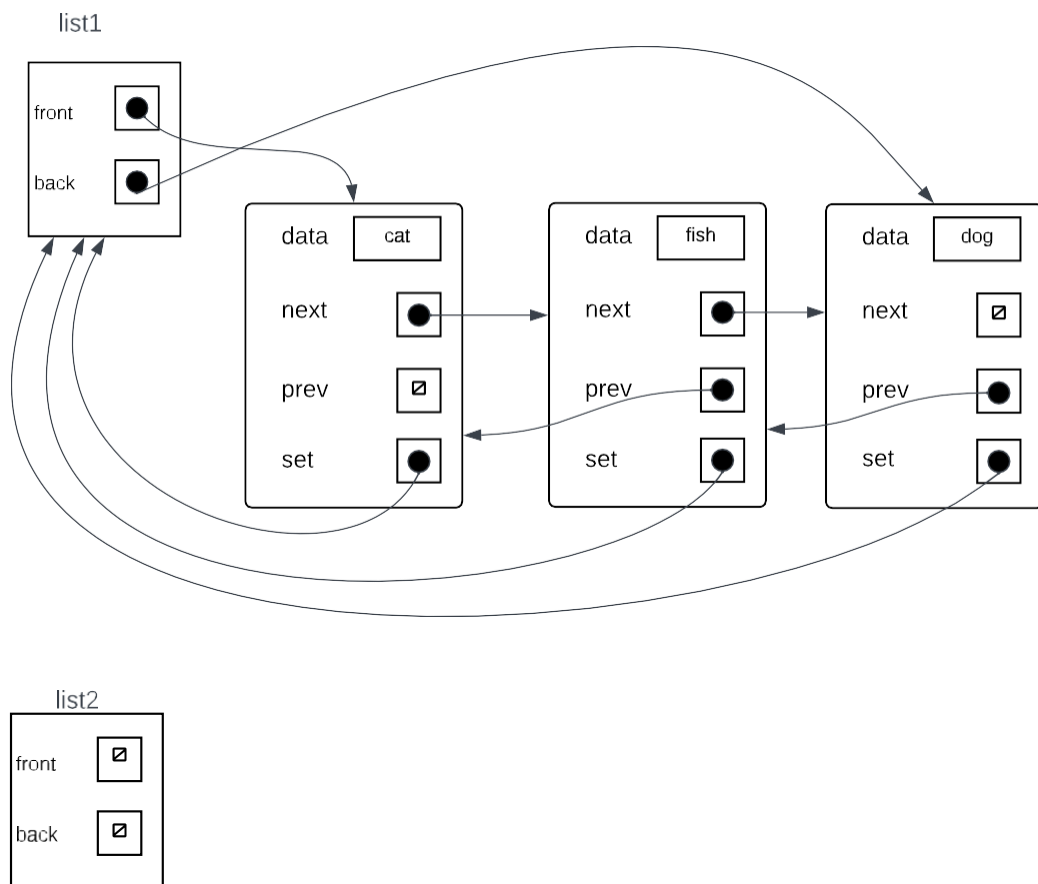What does the function return?

list2.union_set(list1)
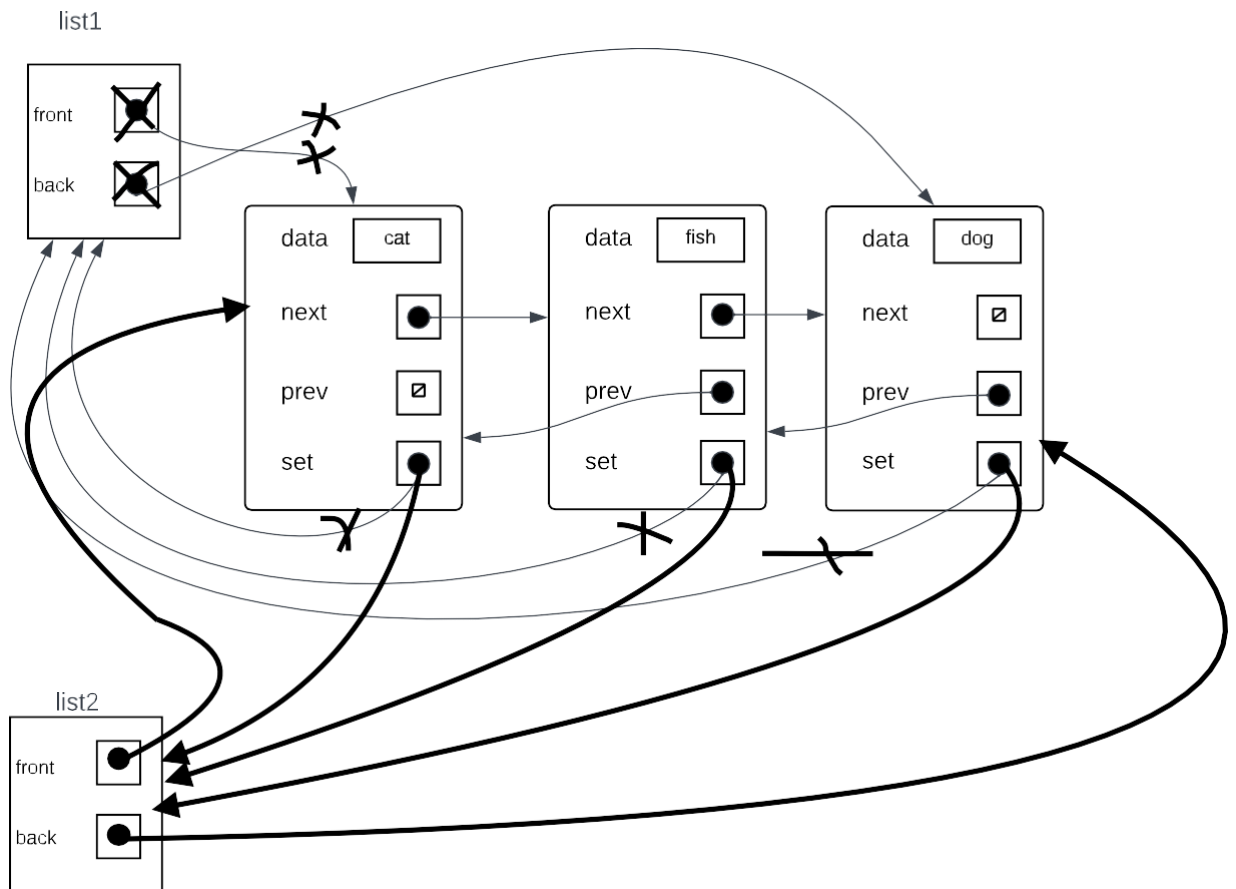
list1

List1's front and back characteristics will be changed to "None." *Union_set()* **will, in accordance with assignment, return the number of nodes transferred, which is equal to list1's length. List1 and List2 will be combined using** *union_set()*. **All 'set' properties on all nodes in list1 will start pointing towards list2 object after** *union_set()*. **List2's front attribute will begin pointing in the direction of list1's front attribute, and list2's back attribute will begin pointing in the direction of list1's back attribute.**

What does the following function call do to these objects?
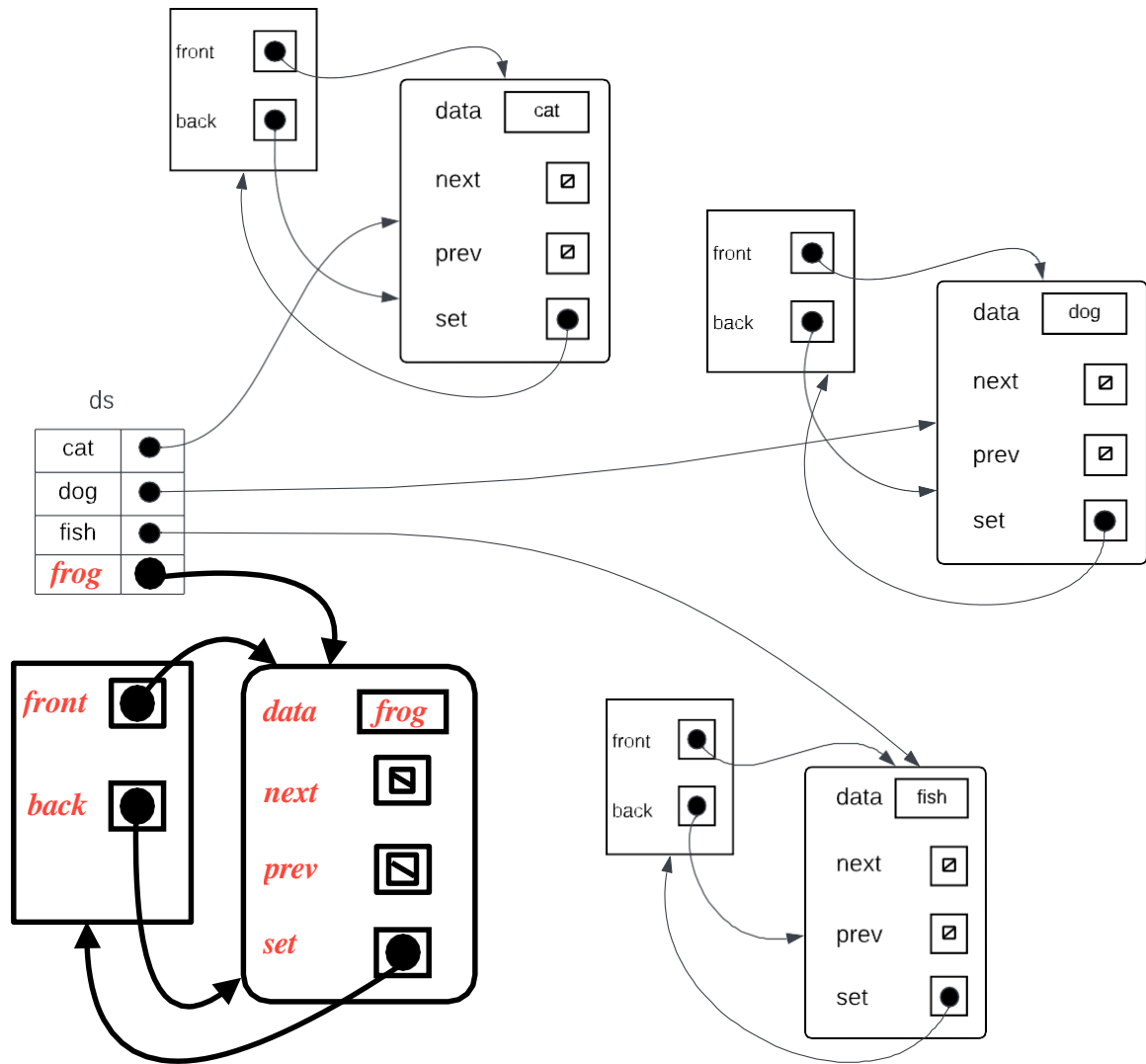What does the function return?

list1.union_set(list2)

list1

front	●
back	●

data	cat
next	●
prev	☒
set	●

data	fish
next	●
prev	●
set	●

data	dog
next	☒
prev	●
set	●

list2

front	☒
back	☒

Since list2 has no nodes, the object structure will remain unchanged.

What does the following function call do to these objects?
What does the function return?

ds.make_set("frog")

Make_set() will first determine whether *"frog"* matches any keys in dictionary "ds," and if it does, it returns false. If not, a new list object will be constructed with a single node that has the "frog" value for the "data" property. *Make_set()* will return True after a new key-value pair with the values *"frog"* and referring to the properly generated node has been generated by the dictionary *"ds"*.

What does the following function call do to these objects?
What does the function return?

ds.union_set("fish","dog")
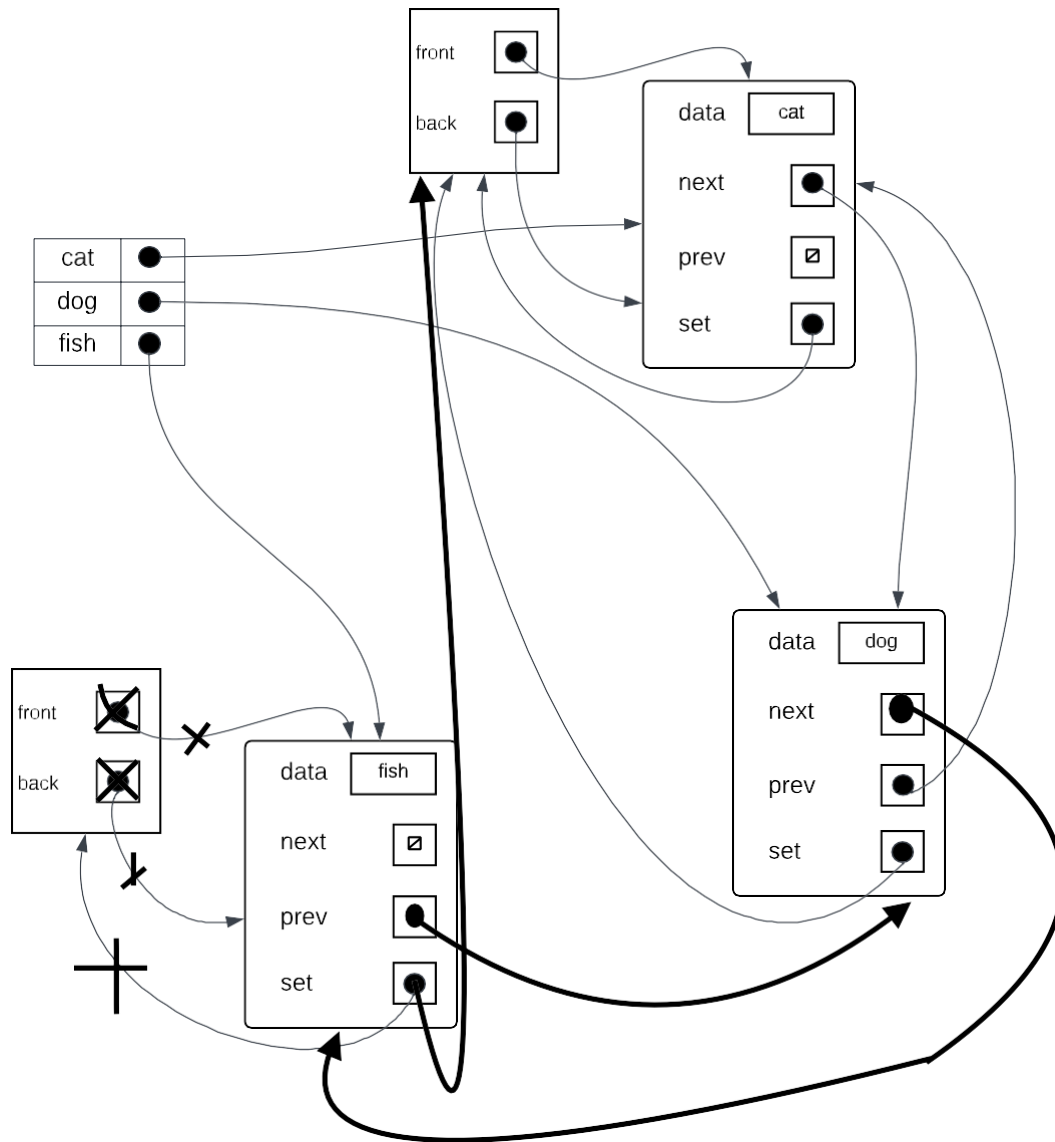
Before returning anything, union_set() will first check to see if "fish" and "dog" are in the dictionary. If one of them isn't or if they're in the same set already, union_set() will return False. As a set with the element "dog" is larger than a set with the item "fish," union_set() will cause the representation of the node with the data "fish" to point in the direction of the node containing the data "dog." The function will then provide a True result.