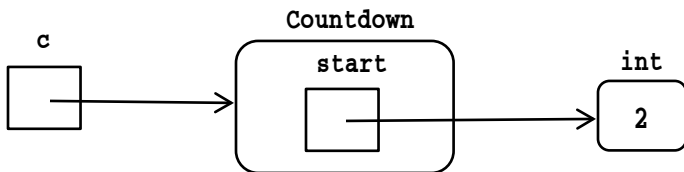
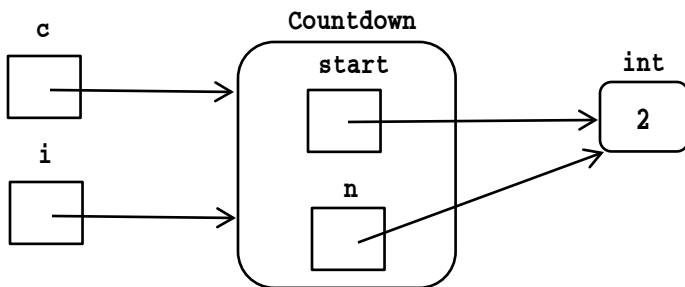


Countdown Iterator: Explicit iter(...) calls

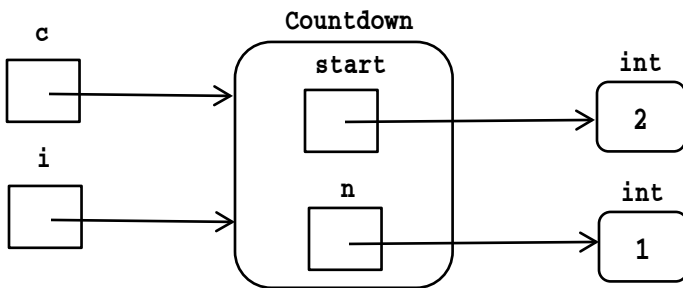
```
c = Countdown(2)
```



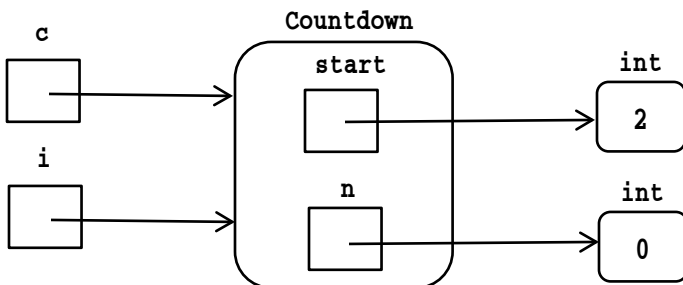
```
i = iter(c)
```



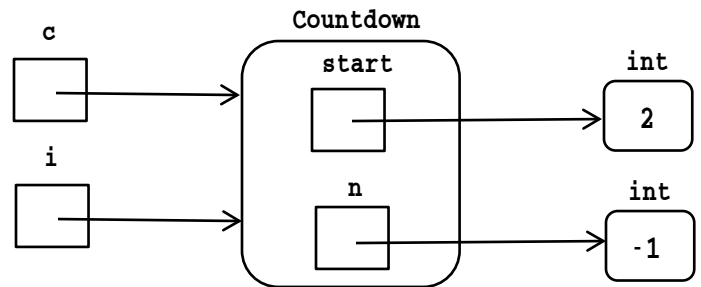
```
print(next(i)) # prints 2
```



```
print(next(i)) # prints 1
```

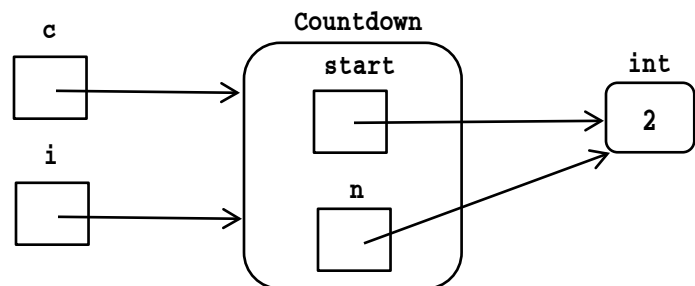


```
print(next(i)) # prints 0
```



```
#next(i) would raise StopIteration
```

```
i = iter(c) # resets n
```



Multiple iterators will share the same **Countdown** object.

Given this behavior, what values would the code on the left print; what values for the code on the right? Hint: how many objects are created on each side?

```
for a in Countdown(2):
```

```
    for b in Countdown(2):
```

```
        print(a,b)
```

```
c = Countdown(2)
```

```
for a in c:
```

```
    for b in c:
```

```
        print(a,b)
```