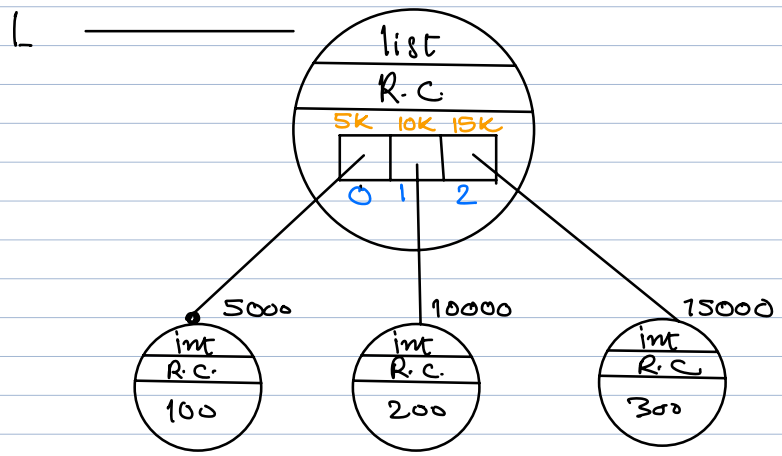
Giving Access Back to individual elements:

L = [100, 200, 300]



L[0] == 100 | L[1] == 200 | L[2] == 300

0 to len(L) - 1

```
i = 0
while i < len(L):
    print (i, L[i])
    i = i+1
```

Principle: Container data types can choose index based or key based access back to a desired individual element (object)

There is another way which involves giving a turn by turn individual access to 'ALL' elements in the container. We cannot access 'a specific' element

of our choice but we can visit all elements in the container.

```
L = [100, 200, 300]
```

```
for x in L:
    print(x)
```

```
for x in [100, 200, 300]:
```

```
for v in iterable:
    Body
```

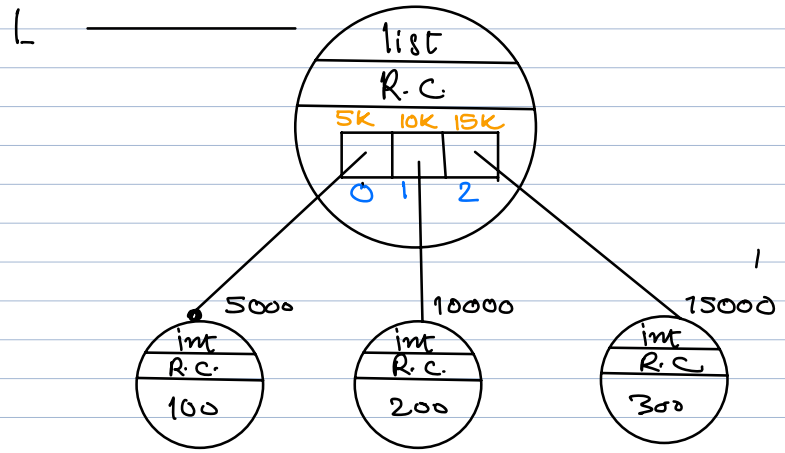
looping statement:
the body of for statement will iterate len(iterable) times.

```
i = 0
while i < len(L):
    Body
    i = i+1
```

```
for v in iterable:
    Body
```

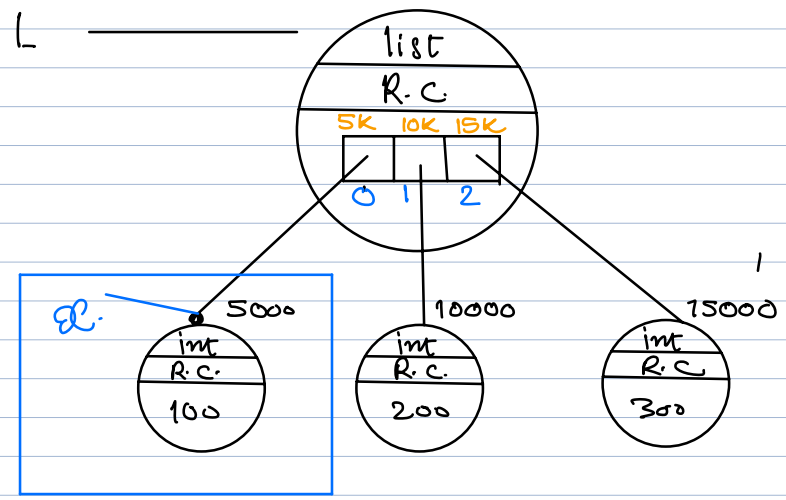
loop variable.

Behaviour of the for loop statement :



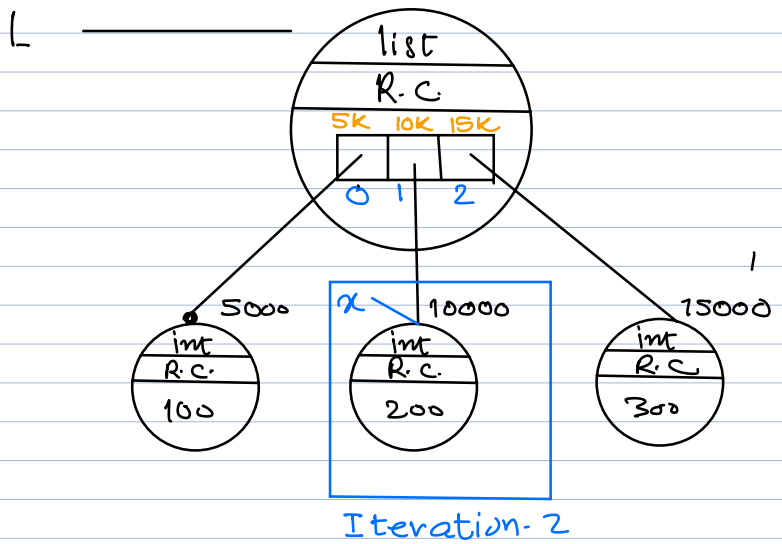
for x in L:	Number of Iterations :
print(x)	len(L) == 3.
	the Body will execute 3 times.

Iteration- 1 :

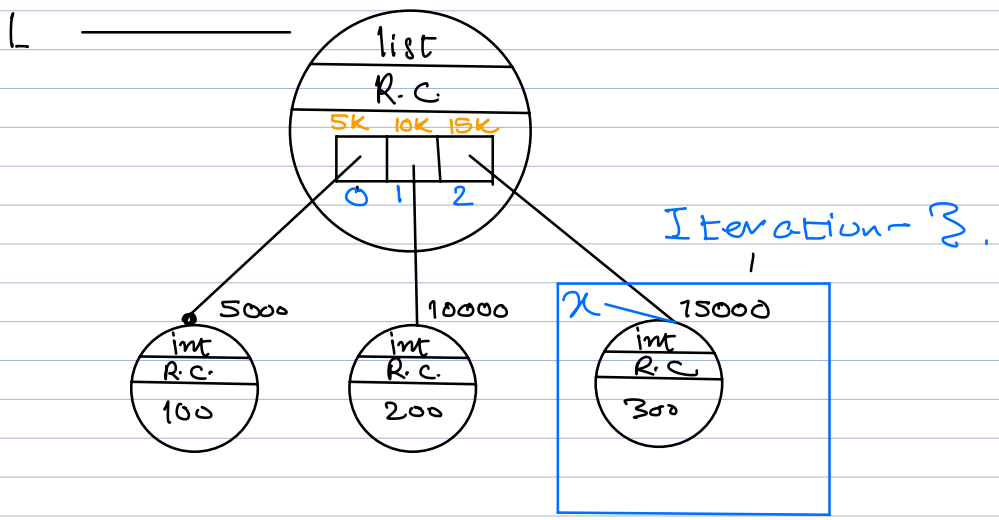


Iteration- 1

print (x) ⟶ 100



print (x) ⟶ 200



print (x) ⟶ 300

```
def my_function(L):
    n = len(L)
```