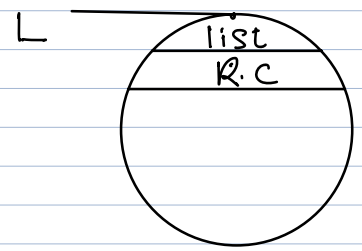
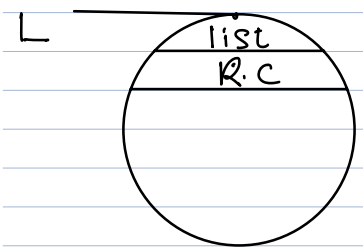
L = [ ]



L.append (10)

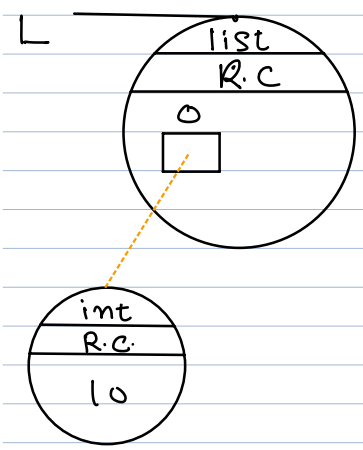
·append() is a function implemented in class list

·It adds new data object in the list at the end position.



• append (10)

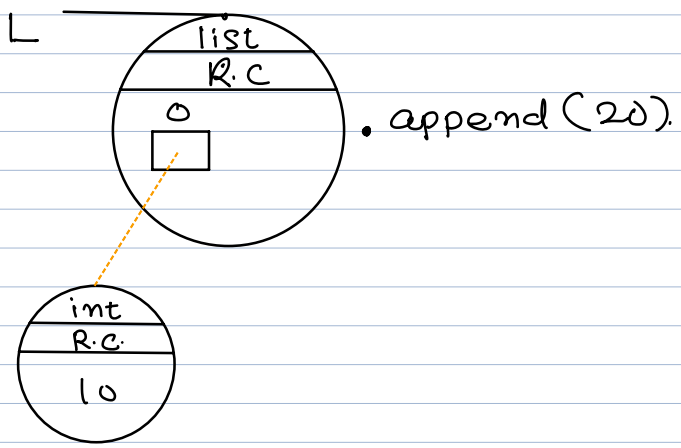
≡



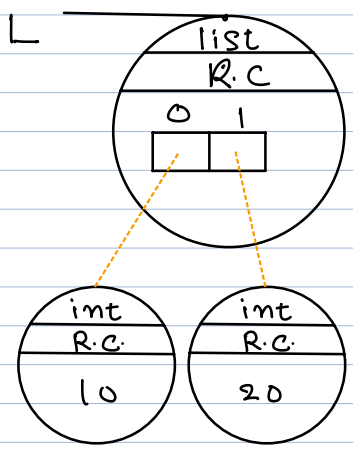
L.append (10) ≡ list.append (L, 10).

[Please refer to the next page]

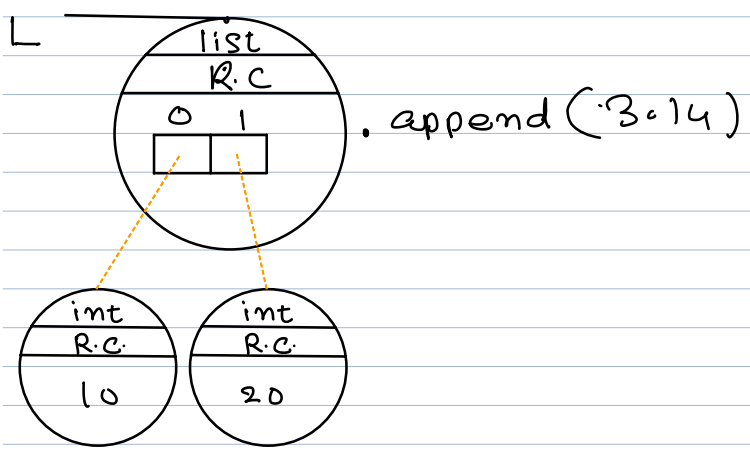
L.append(20) ≡ list.append(L, 20)

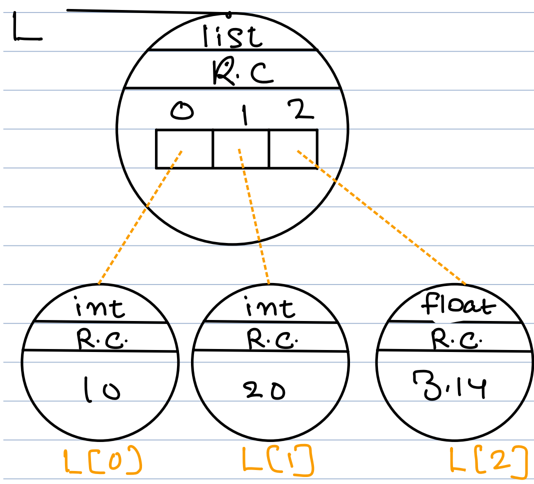


≡



L.append(3.14) ≡ list.append(L, 3.14)

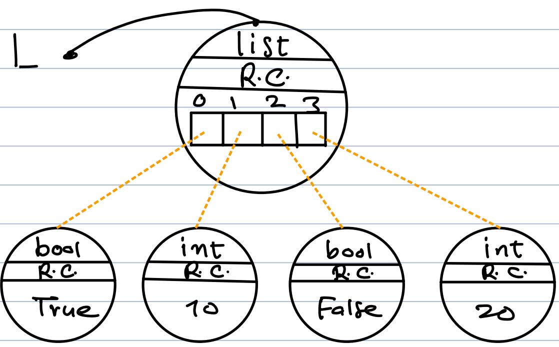




Question: How to travel through the list (or how to visit each element in the list) using while loop statement?

```
[
i = 0          → loop variable initialization.
while i < 5:   → loop variable condition.
   i=i+1       → loop variable modification.
]
```

Example list: L = [True, 10, False, 20].



```
>>> print (L[0])
    True
>>> print (L[1])
    10
>>> print (L[2])
    False
>>> print (L[3])
    20
```

Problem-1: What if the length of the list is considerably large? say, 5000 [Inconvenience]

Problem-2: What if the length of the list is known at the run time and not at the time of writing the code.

Main Problem: How to use while loop in order to visit and print all elements in the list.

↳ Subproblem: Given the list L, write a while loop to generate all its valid indices.

Valid index range for list L is: 0 to (length of L − 1).

```
>>len(L)
```

(total number of elements in the list).

0 to len(L)−1 is a valid index range for list named L.

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

VERY ADVANCED. BEGINNERS SHOULD SKIP.

[on 32 bit machine :

4 GB - Size of virtual address space of Kernel

- Size of virtual address space occupied by the Python interpreter - memory occupied by all objects in Python except the list object]

[on 64 bit machine :

256 TiB - Size of virtual address space of Kernel

- Size of virtual address space occupied by the Python interpreter - memory occupied by all objects in Python except the list object]

Size of virtual address space of Kernel

= 1GB for Linux | 2GB for windows.

END OF ADVANCED SECTION.

Solution to main problem:

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