Operatori pt liste O(len(L))

+, +=, * (in, not in, relationali)

L = [1,2] + [3,4,5] => L = [1,2,3,4,5]

L = [1,2] * 3 = [1,2,1,2,1,2]

[1,2,6,5] > [1,2,3]

[1,2,6,"Pop Ion"] > [1,2,4,3,14]

[1, 2, 3.14, "Pop Ion", I] > [1,2, Mu...

?!

Funcții predefinite pt. liste:

a) len(liste)

len([1,2,[3,4,5]]) = 3

b) list(secvență)

list("test") = ["t","e","s","t"]

c) min(liste)/max(liste)

d) sum(liste)

print(sum([int(x) for x in input("...

e) sorted(secvență)

L = sorted("test") = ["e","s","t","t"]

s = "".join(L) s = "estt"

L1 = [5,1,7,3]

L2 = sorted(L1) => L2 = [1,3,5,7]

Metode din clasa list

a) count(valoare)

b) append(valoare)

```
L=[1,2]
L.append([3,4,5]) => L=[1,2,[3,4,5]]
```

c) extend(secventa)

```
L=[1,2]
L.extend([3,4,5]) => L=[1,2,3,4,5]
L.extend([3,[4,5]]) => L=[1,2,3,[4,5]]
```

d) insert(pozitie, valoare)

```
L=[1,2,3,4]
L.insert(2,6) => [1,2,6,3,4]
```

e) index(valoare)

```
L=[1,2,3,4]
n=L.index(5) => EROARE!
```

Varianta 1:

```
if x in L:
    n=L.index(x); print(n)
else:
    print("nu apare")
```

Varianta 2:

```
try:
    n=L.index(x)
    print(n)
except: VALUE_ERROR
    print("NU APARE")
```

f) remove (valeur)

g) pop ([position]) → indexError

L = [1, 2, 3, 4]

x = L.pop(2)

print(L) => [1, 2, 4]

h) .clear() => = []

i) .sort()

L = [1, 5, 7, 2]

L.sort() => L = [1, 2, 5, 7]