

<https://www.hackerrank.com/challenges/python-lists/problem?isFullScreen=false>

```
list1=[]
n=int(input())
for i in range(n):
    cmd=input().split()
    if cmd[0]=="append":
        list1.append(int(cmd[1]))
    elif cmd[0]=="insert":
        list1.insert(int(cmd[1]),int(cmd[2]))
    elif cmd[0]=="remove":
        list1.remove(int(cmd[1]))
    elif cmd[0]=="pop":
        list1.pop()
    elif cmd[0]=="reverse":
        list1.reverse()
    elif cmd[0]=="print":
        print(list1)
    elif cmd[0]=="sort":
        list1.sort()
```

copy()

This method creates copy of the list.

Python support two types of copies

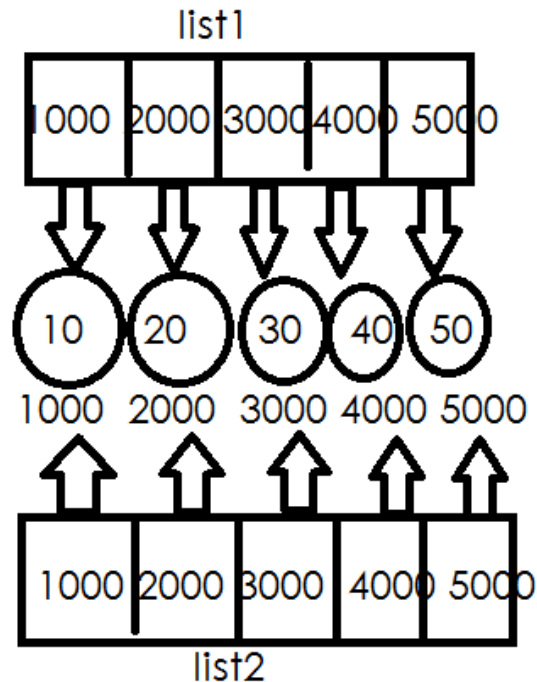
1. Shallow copy
2. Deep copy

copy() method of list create shallow copy of the list. the shallow copy is called reference copy.

Shallow copy

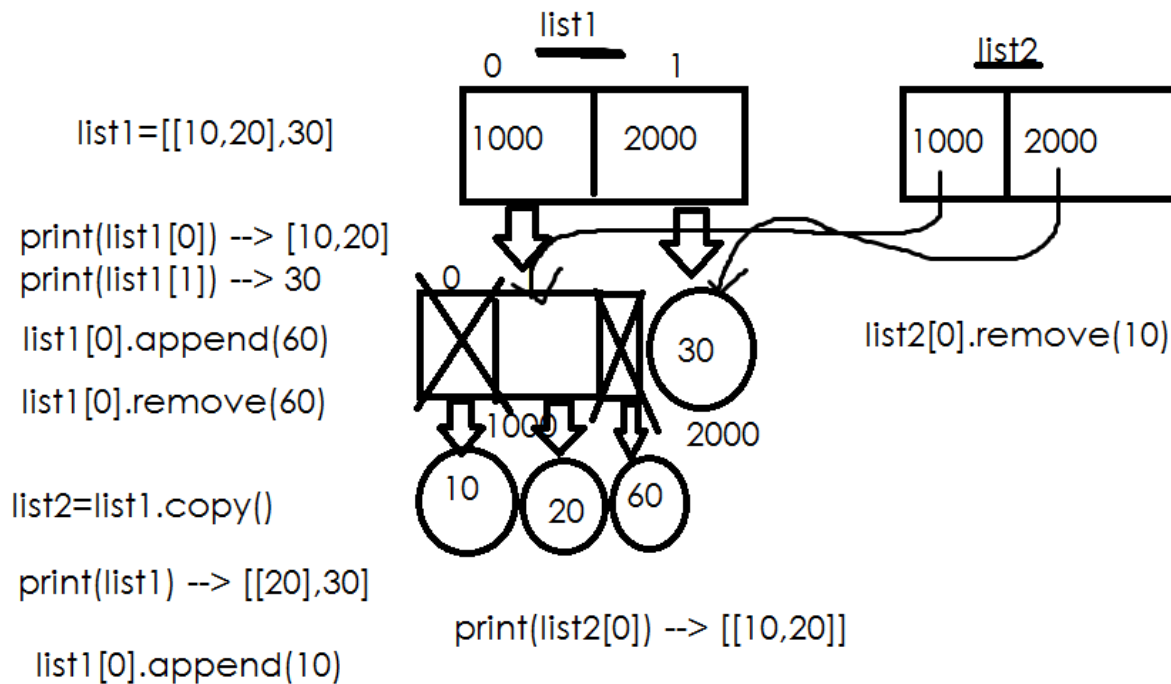
```
list1=[10,20,30,40,50]
```

```
list2=list1.copy()
```



Example:

```
>>> list1=[10,20,30,40,50]
>>> list2=list1.copy()
>>> print(list1)
[10, 20, 30, 40, 50]
>>> print(list2)
[10, 20, 30, 40, 50]
>>> print(id(list1[0]))
1021623206416
>>> print(id(list2[0]))
1021623206416
```



```
>>> l1=[[10,20],30]
>>> l2=l1.copy()
>>> print(l1)
[[10, 20], 30]
>>> print(l2)
[[10, 20], 30]
>>> l1[0].append(40)
>>> print(l1)
[[10, 20, 40], 30]
>>> print(l2)
[[10, 20, 40], 30]
>>> l2[0].remove(10)
>>> print(l1)
[[20, 40], 30]
>>> print(l2)
[[20, 40], 30]
```

Deep copy

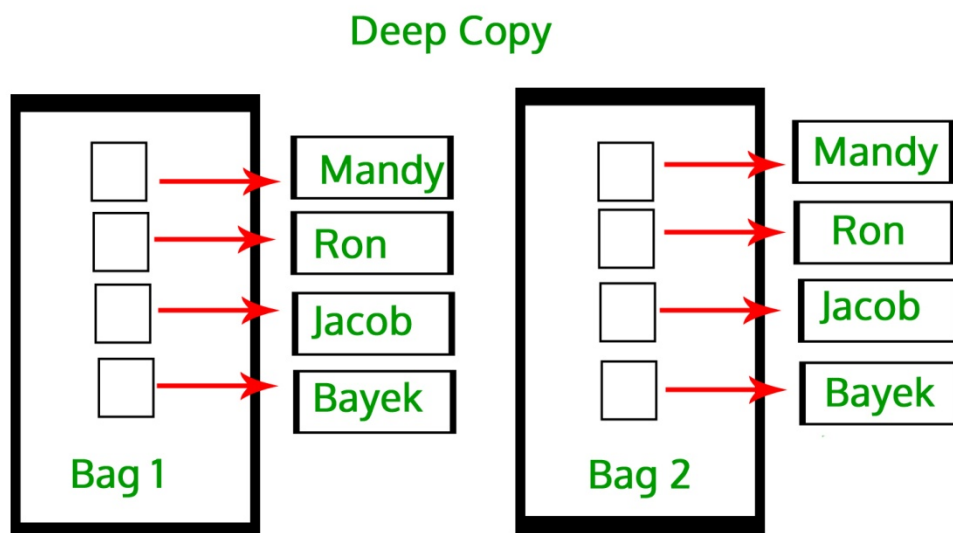
A shallow copy constructs a new compound object and then (to the extent possible) inserts references into it to the objects found in the original. A deep copy constructs a new compound object and then, recursively, inserts copies into it of the objects found in the original

Deep copy is object copy. It will create copy of object and insert into new list.

This deep copy is done using deepcopy function provided by “copy” module of python. “copy” is a predefined module which comes with python software.

In deep copy any changes done in one list, that changes does not reflect to another list.

In shallow copy any changes done in one list that changes reflect to another list.



Example:

```
>>> list1=[[10,20],30]
>>> import copy
>>> list2=copy.deepcopy(list1)
>>> print(list1)
```

```

[[10, 20], 30]
>>> print(list2)
[[10, 20], 30]
>>> list1[0].append(40)
>>> print(list1)
[[10, 20, 40], 30]
>>> print(list2)
[[10, 20], 30]
>>> list1[0].remove(40)
>>> print(list1)
[[10, 20], 30]
>>> print(list2)
[[10, 20], 30]

```

Replace values of list

We can replace the values of list, this replacing is done using,

1. Index
2. Slicing

Using index we can replace one value

Using slicing we can replace more than one value

Example:

```

list1=[10,20,30,40,50]
print(list1)
[10, 20, 30, 40, 50]
>>> list1[0]=99
>>> print(list1)
[99, 20, 30, 40, 50]
>>> list1[-1]=77
>>> print(list1)
[99, 20, 30, 40, 77]
>>> list1[9]=100

```

Traceback (most recent call last):

File "<pyshell#34>", line 1, in <module>

list1[9]=100

IndexError: list assignment index out of range

Example:

```

>>> list1=list(range(10,110,10))
>> print(list1)
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list1[0:2]=[11,22]
>>> print(list1)
[11, 22, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list1[-2:]=[99,111]
>>> print(list1)
[11, 22, 30, 40, 50, 60, 70, 80, 99, 111]
>>> list1[-1:-3:-1]=[100,90]
>>> print(list1)
[11, 22, 30, 40, 50, 60, 70, 80, 90, 100]
>>> list1[3:6]=[44,55,66]
>>> print(list1)
[11, 22, 30, 44, 55, 66, 70, 80, 90, 100]

```

4 3 5 2 1
 3 4 5 2 1
 3 4 5 2 1
 3 4 2 5 1
 3 4 2 1 5

```

for i in range(n):
    for j in range(0,n-1)
        if list1[j]>list1[j+1]:
            list1[j],list1[j+1]=list1[j+1],list1[j]

```

Example:

write a program to sort the elements of list
 # in ascending order using bubble sort

```

list1=[]
n=int(input("enter how many element"))

```

```
for i in range(n):
    value=int(input("enter value"))
    list1.append(value)
print("Before sorting ",list1)
for i in range(n):
    for j in range(n-1):
        if list1[j]>list1[j+1]:
            list1[j],list1[j+1]=list1[j+1],list1[j]

print("after sorting ",list1)
```

Output:

```
enter how many element5
enter value4
enter value3
enter value5
enter value2
enter value1
Before sorting [4, 3, 5, 2, 1]
after sorting [1, 2, 3, 4, 5]
```

Nested List