

N_1

obj.join(lst) --ეს ფუნქცია იღებს ორ არგუმენტს ერთი არის lst --(ლისტი ან ტუპლე ტიპის ცვლადი) და მეორე obj--(სტრინგის ობიექტი ან ასო-სიმბოლო) და აბრუნებს lst--ში შემავალი ელემენტების გაერთიანებას obj--ობიექტით

მაგ:

```
1  ls=["ალექსანდრე","მიშკო","დათო","ალექსანდრე","საბა","ბექა"]
2  value=" _ <3 _ ".join(ls)
3  cvladis_tipi=type(value)
4  print(value)
5  print(cvladis_tipi)
```

ალექსანდრე _ <3 _ მიშკო _ <3 _ დათო _ <3 _ ალექსანდრე _ <3 _ საბა _ <3 _ ბექა
<class 'str'>
[Finished in 0.1s]

დაწერეთ my_join(lst,obj) ფუნქცია

N_2

list.insert(index, obj) --ეს ფუნქცია იღებს სამ არგუმენტს list--ლისტს ,index--ინდექსს და obj--ობიექტს, ფუნქცია lst--ლისტში , obj --ობიექტს ჩასვავს index--ის ტოლ ადგილას

მაგ:

```
1  ls=["ალექსანდრე","მიშკო","დათო","ალექსანდრე","საბა","ბექა"]
2  ls.insert(3,"ცოტნე")
3  print(ls)
```

['ალექსანდრე', 'მიშკო', 'დათო', 'ცოტნე', 'ალექსანდრე', 'საბა', 'ბექა']
[Finished in 0.2s]

დაწერეთ my_insert(lst,obj,index) ფუნქცია

N_3

list.pop() ეს ფუნქცია list--ლისტს მოაშორებს ბოლო ელემენტს და დაგვიბრუნებს , ანუ მას დეფაულტად აქვს ბოლო ელემენტის ინდექსი (-1) , ხოლო თუ მას გადავცემთ იმ ელემენტის ინდექს რომელიც გვინდა რომ მოაშოროს და დაგვიბრუნოს ფუნქციას ექნება ასეთი სახე **list.pop(index)**

მაგ_1:

```
1 ls=["ალექსანდრე","მიშიკო","დათო","ალექსანდრე","საბა","ბექა"]
2 obj=ls.pop()
3 print(ls)
4 print(obj)

['ალექსანდრე', 'მიშიკო', 'დათო', 'ალექსანდრე', 'საბა']
ბექა
[Finished in 0.1s]
```

მაგ_2:

```
1 ls=["ალექსანდრე","მიშიკო","დათო","ალექსანდრე","საბა","ბექა"]
2 obj=ls.pop(2)
3 print(ls)
4 print(obj)

['ალექსანდრე', 'მიშიკო', 'ალექსანდრე', 'საბა', 'ბექა']
დათო
[Finished in 0.1s]
```

დაწერეთ my_pop(lst,index=-1) , გამოიყენეთ დეფაულტი არგუმენტის მეთოდი... , ობიექტის წასაშლელად გამოიყენეთ del lst[index] მეთოდი *(შეეცადეთ დაწეროთ del –მეთოდის გამოყენების გარეშე ბონუს ქულების მსურველებმა)

N_4

list.remove(obj) --ეს ფუნქცია list--ლისტში წაშლის obj--ობიექტს

მაგ:

```
1 ls=["ალექსანდრე","მიშიკო","დათო","ალექსანდრე","საბა","ბექა"]
2 ls.remove("დათო")
3 print(ls)

['ალექსანდრე', 'მიშიკო', 'ალექსანდრე', 'საბა', 'ბექა']
[Finished in 0.1s]
```

დაწერეთ my_remove(lst,obj) --ფუნქცია , შეგიძლიათ გამოიყენოთ უკვე გავლილი ფუნქციები

N_5

list.reverse() --ეს ფუნქცია --ლისტს ატრიალებს

მაგ :

```
1 ls=["ალექსანდრე","მიშიკო","დათო","ალექსანდრე","საბა","ბექა"]
2 ls.reverse()
3 print(ls)

['ბექა', 'საბა', 'ალექსანდრე', 'დათო', 'მიშიკო', 'ალექსანდრე']
[Finished in 0.1s]
```

დაწერეთ my_reverse(lst) -- ფუნქცია

N_6

list.sort() --ეს ფუნქცია --ლისტის ობიექტებს ალაგებს ზრდადობის მიხედვით

მაგ:

```
1 ls=["ალექსანდრე","მიშიკო","დათო","ალექსანდრე","საბა","ბექა"]
2 ls.sort()
3 print(ls)

['ალექსანდრე', 'ალექსანდრე', 'ბექა', 'დათო', 'მიშიკო', 'საბა']
[Finished in 0.1s]
```

აქ დალაგდა ანბანის მიხედვით ...

დაწერეთ my_sort(lst)--ფუნქცია , (დაგუგლეთ bubble sort ;))

დამატებითი ინფორმაცია:

Sort() მეთოდს აქვს სხვა და სხვა სახეები , იხილეთ : <http://pythoncentral.io/how-to-sort-a-list-tuple-or-object-with-sorted-in-python/>

შემდეგ გაკვეთილზე დავწერთ თამაშს კრისწი-ნოლიკს , იფიქრეთ მის ალგორითმზე