6. PYTHONDA MA’LUMOT TO’PLAMLARI VA TURLARI

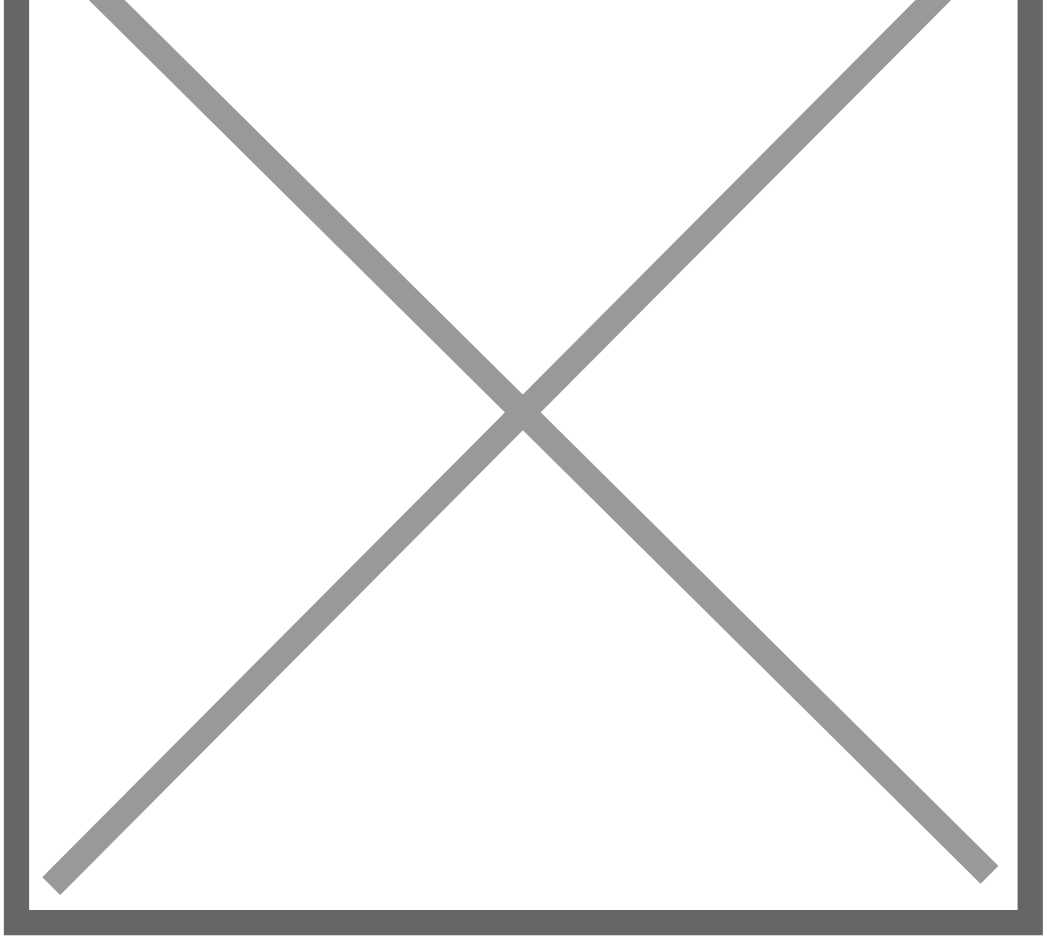


RO’YXAT(LIST) BILAN ishlash

Avvalgi mavzuimizda biz o'zgaruvchi yaratish, va uning ichida biror qiymatni (matn yoki son) saqlashni o'rgandik. Bunda biz bitta o'zgaruvchiga bitta qiymat berdik xolos.

Bugun o'rganadigan navbatdagi mal'umot turi Ro’yxar (List) deb ataladi. Ro'yxat o'z nomi bilan, bitta o'zgaruvchida bir nechta qiymatlarni saqlash imkonini beradi. Bu qiymatlar Ro’yxar (List) **elementlari** deyiladi. Ro'yxatda son, matn yoki aralash turdagi elementlarni saqlash mumkin.

Ro’yxar (List) quyidagicha yaratiladi:

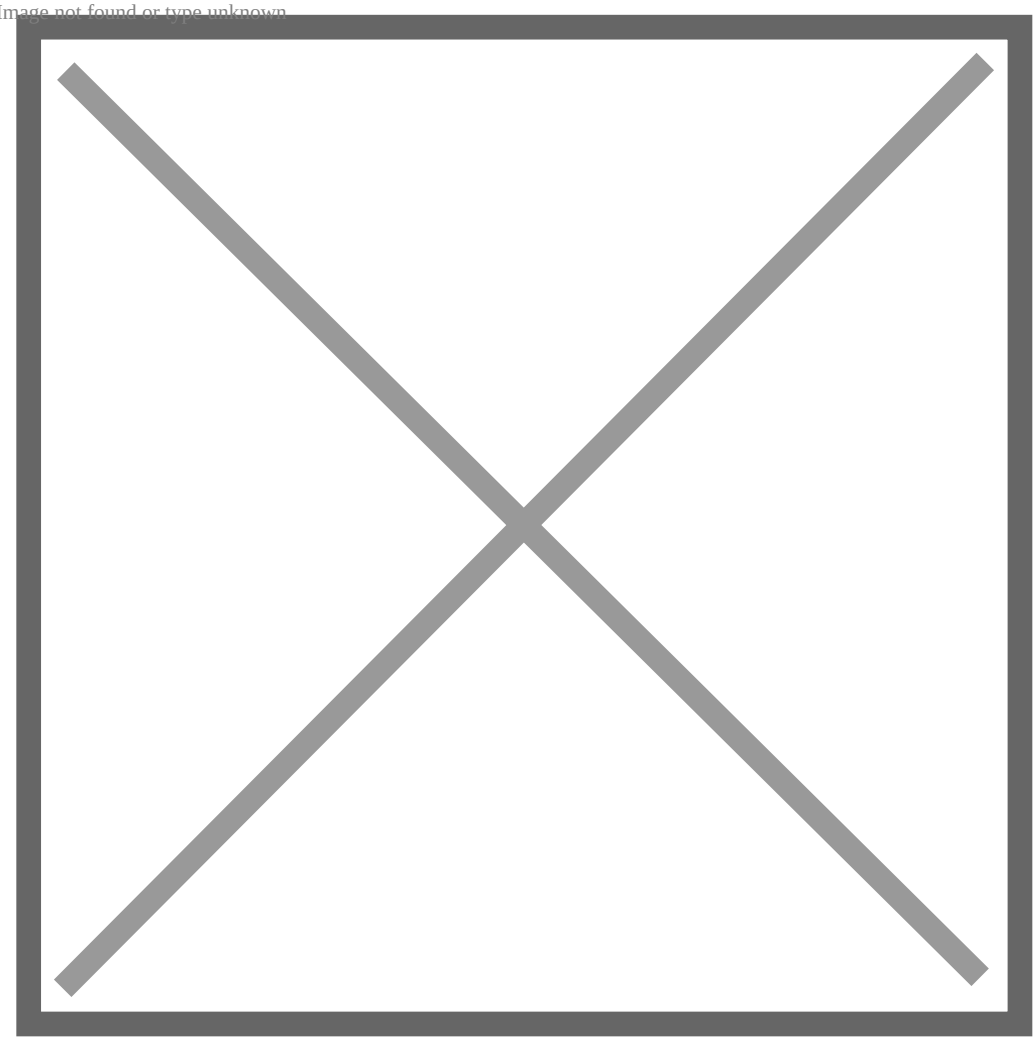


LIST

ELEMENTLARI

Ro'yxatdagi har bir element tartib bilan joylashgani sababli, biz istalgan elementga uning tartib raqami (indeksi) bo'yicha murojat qilishimiz mumkin. Listning birinchi elementi 0 dan boshlanadi, ikkinchisi 1 deb davom etadi va hokazo.

Image not found or type unknown



Natija:

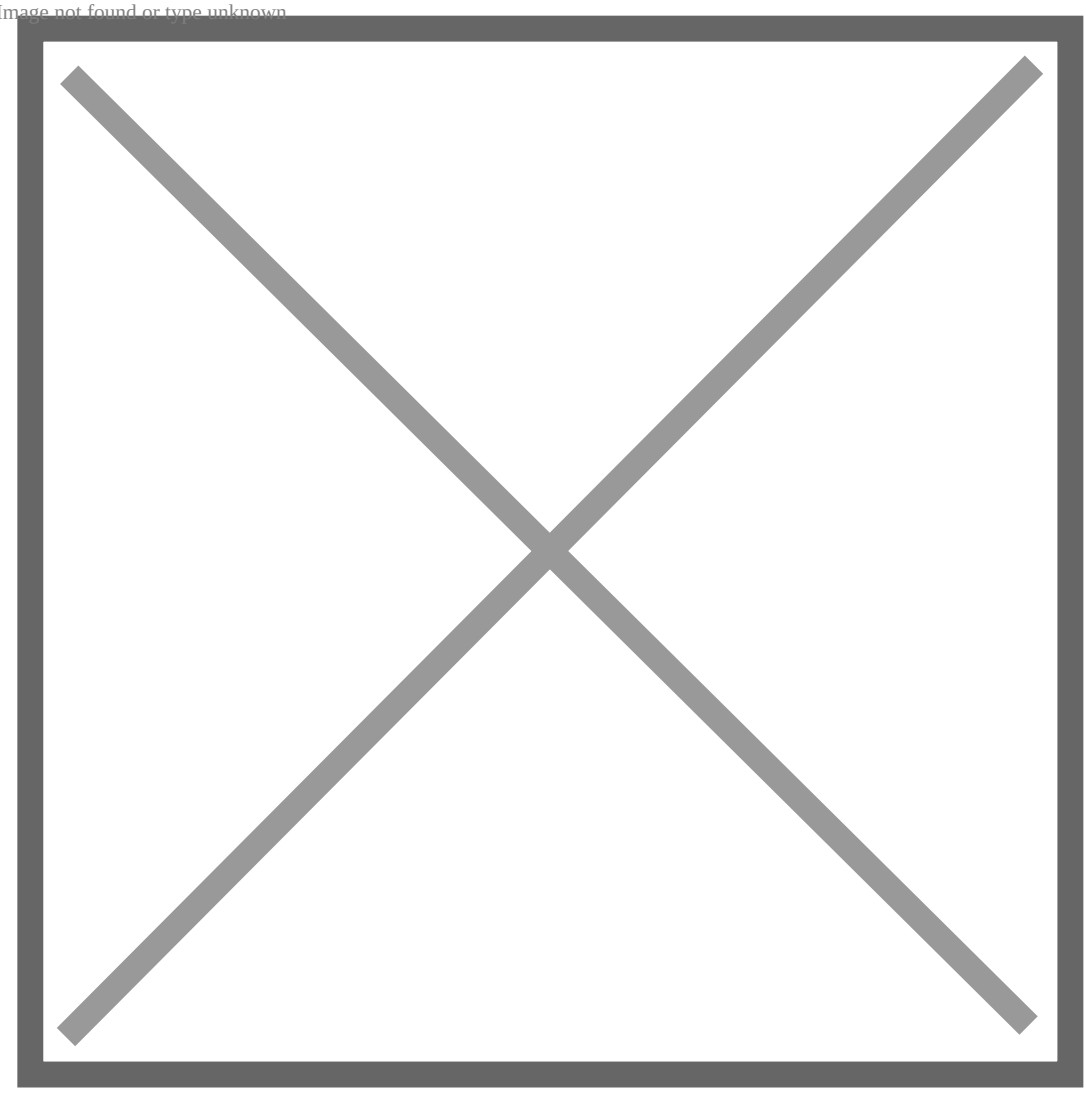
Birinchi gul: atirgul

Ikkinchi butun son: 478

Agar List elementlari matn bo’sa, matn metodlarini qo’llashingiz mumkin. Listning oxirgi elementi -1 hisoblanadi.

1. List elementlari ustida arifmetik amallar bajarish:

Image not found or type unknown



Natija: 1283

2. Dastur davomida listning tarkibi o'zgarishi, yangi elementlar qo'shilishi, ba'zi elementlar o'chirilishi tabiiy hol. Ro'yxatdagi biror elementning qiymatini o'zgartirish uchun, o'sha elementga indeksi bo'yicha murojat qilamiz va yangi qiymat yuklaymiz.

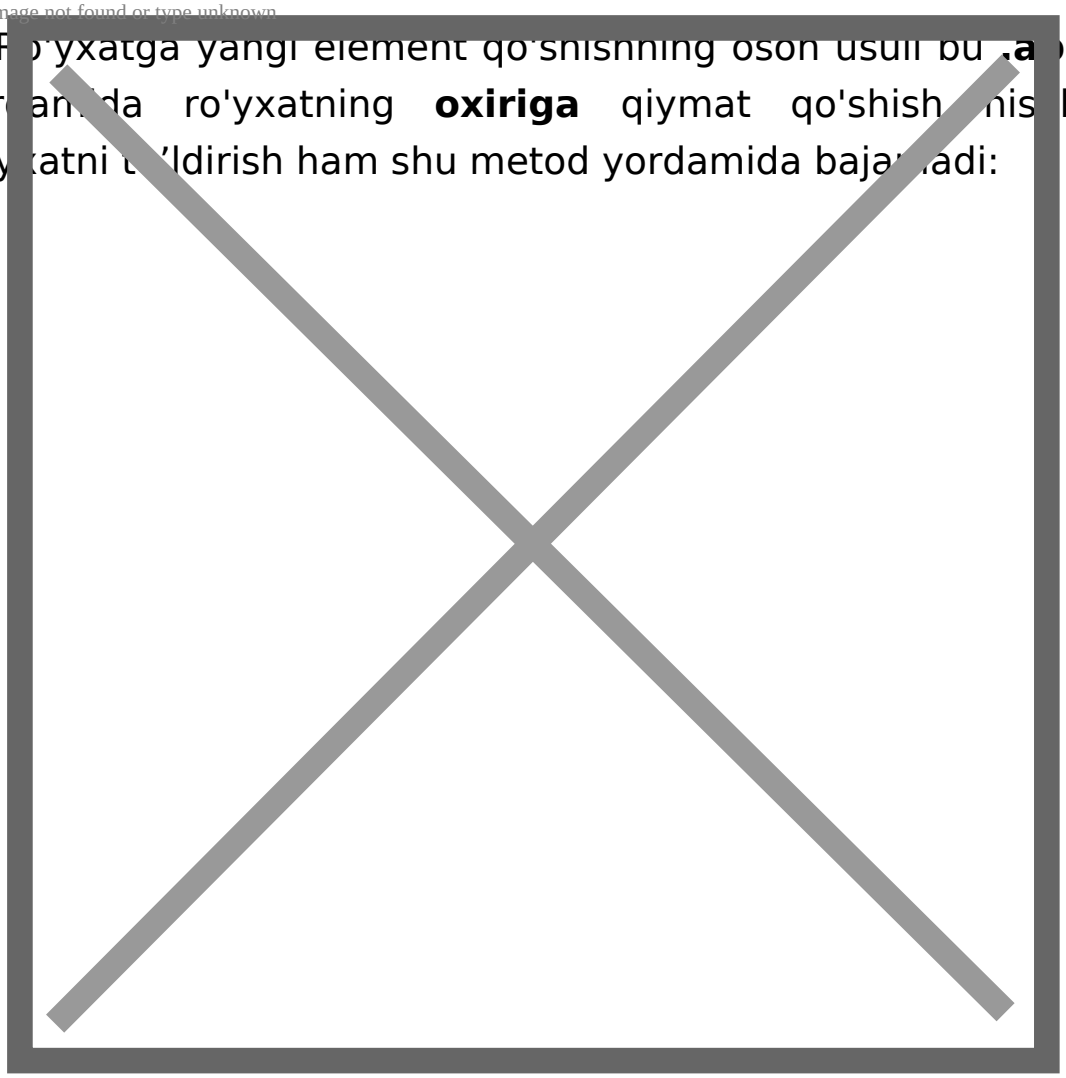
Image not found or type unknown



Natija: [12, 71, 87, 74, 9]

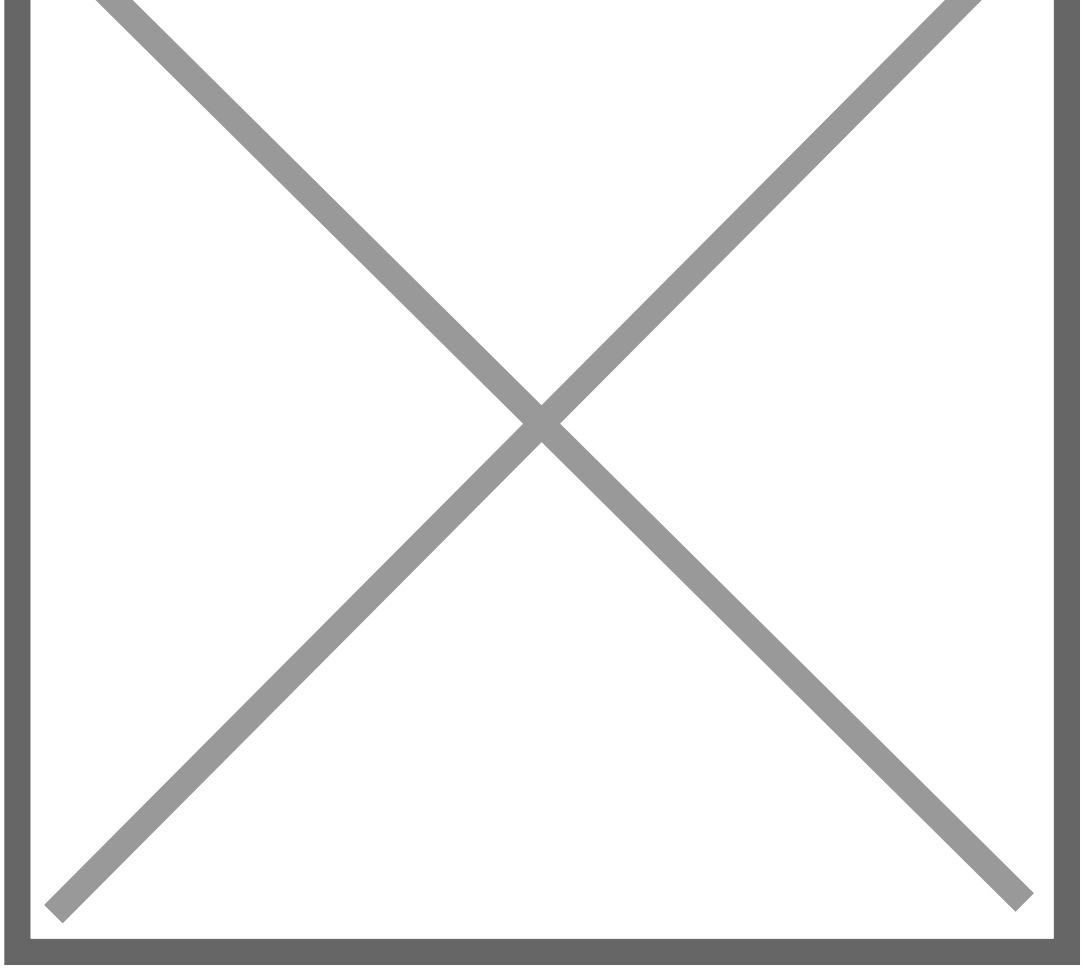
Image not found or type unknown

3. Ro'yxatga yangi element qo'shishning oson usuli bu **.append()** metodi yordamida ro'yxatning **oxiriga** qiymat qo'shish amalga oshiriladi, bo’sh ro’yxatni to’ldirish ham shu metod yordamida bajariladi:



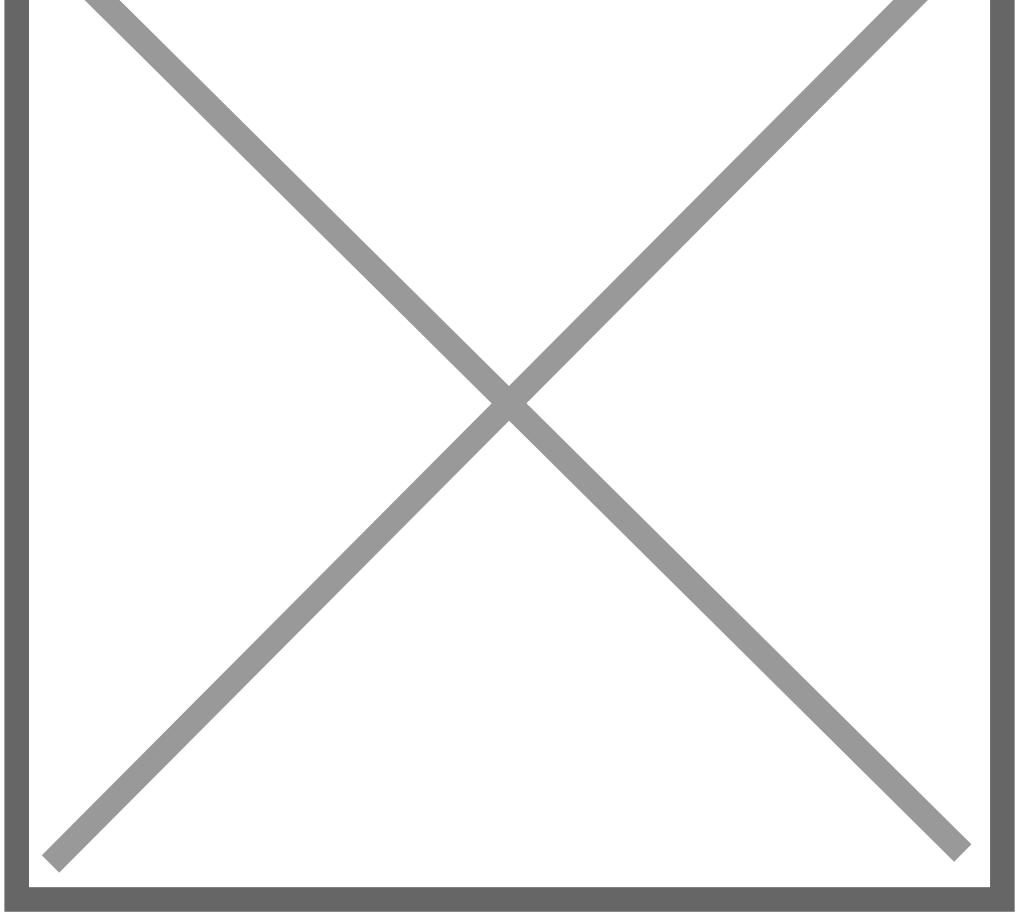
Natija: ['bir', 'bech', 87, 74, 9, 'etti', 49]

4. Ro'yxatning istalgan joyiga yangi element qo'shish uchun **.insert()** metodidan foydalanamiz. .insert() metodi ichida yangi elementning indeksi va qiymati beriladi:



Natija: ['ikki', 'bir', 'bech', 789, 87, 54, 74, 9]

5. Ro'yxatdan biror elementni olib tashlash uchun uning indeksini yoki qiymatini bilishimiz lozim. Inedeks yordamida olib tashlash uchun **del** operatoridan, element qiymati bo'yichi o'chirish uchun esa **.remove(qiymat)** metodidan foydalanamiz:



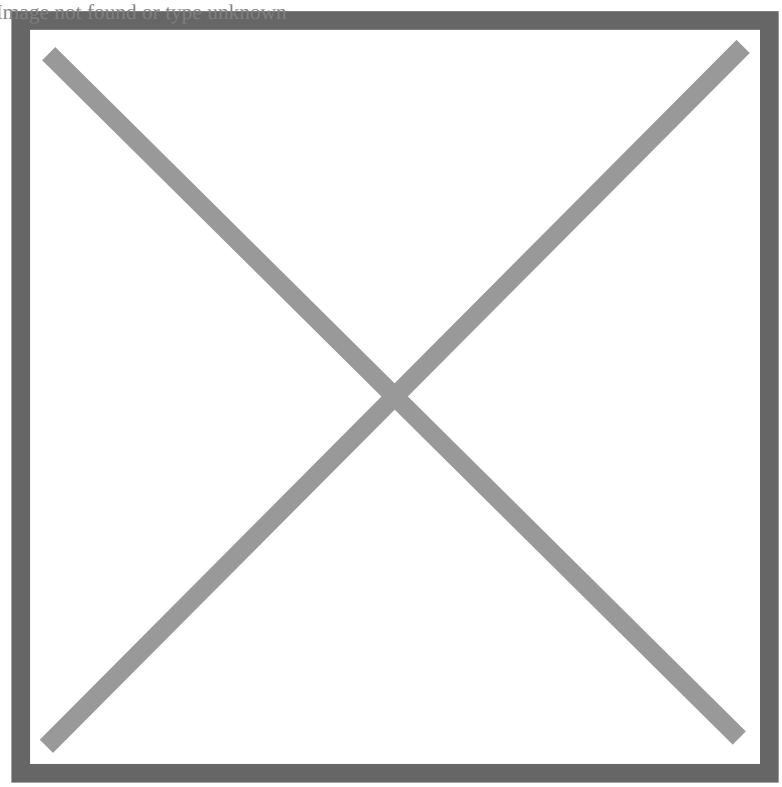
Natija:

['Salim', 'Sobir', 'Karim']

['Anvar', 'Salim', 'Karim']

6. Ro’yxat elementlari ichidan borortasidan foydalanish uchun Pythonda **.pop(indeks)** metodidan foydalaniladi.

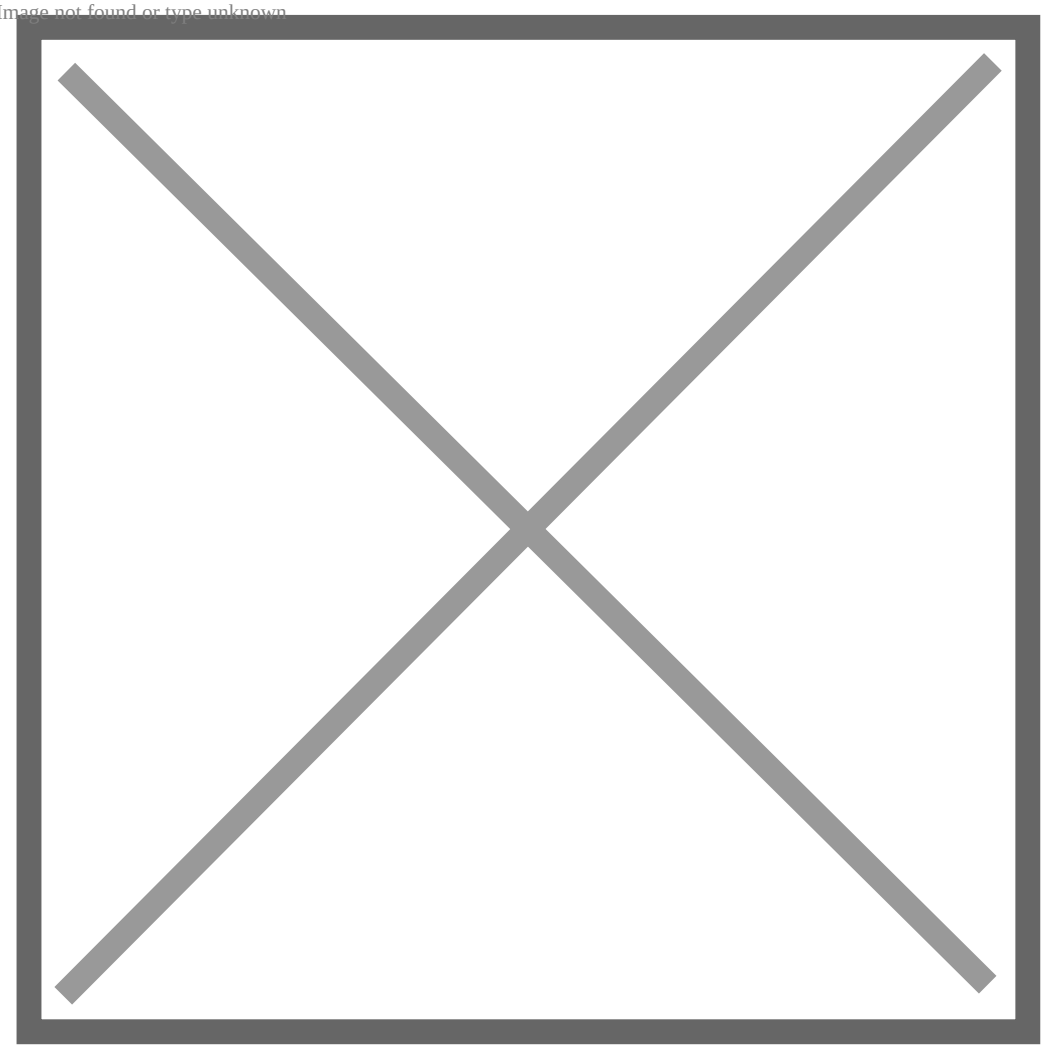
Image not found or type unknown



Natija: Bizning faxrimiz alochi talaba Karim

7. Aksar holatlarda ro'yxat ichidagi elementlarni alifbo ketma-ketligida tartiblash talab qilinishi mumkin. Buning uchun list uchun maxsus **.sort()** metodidan foydalanamiz.

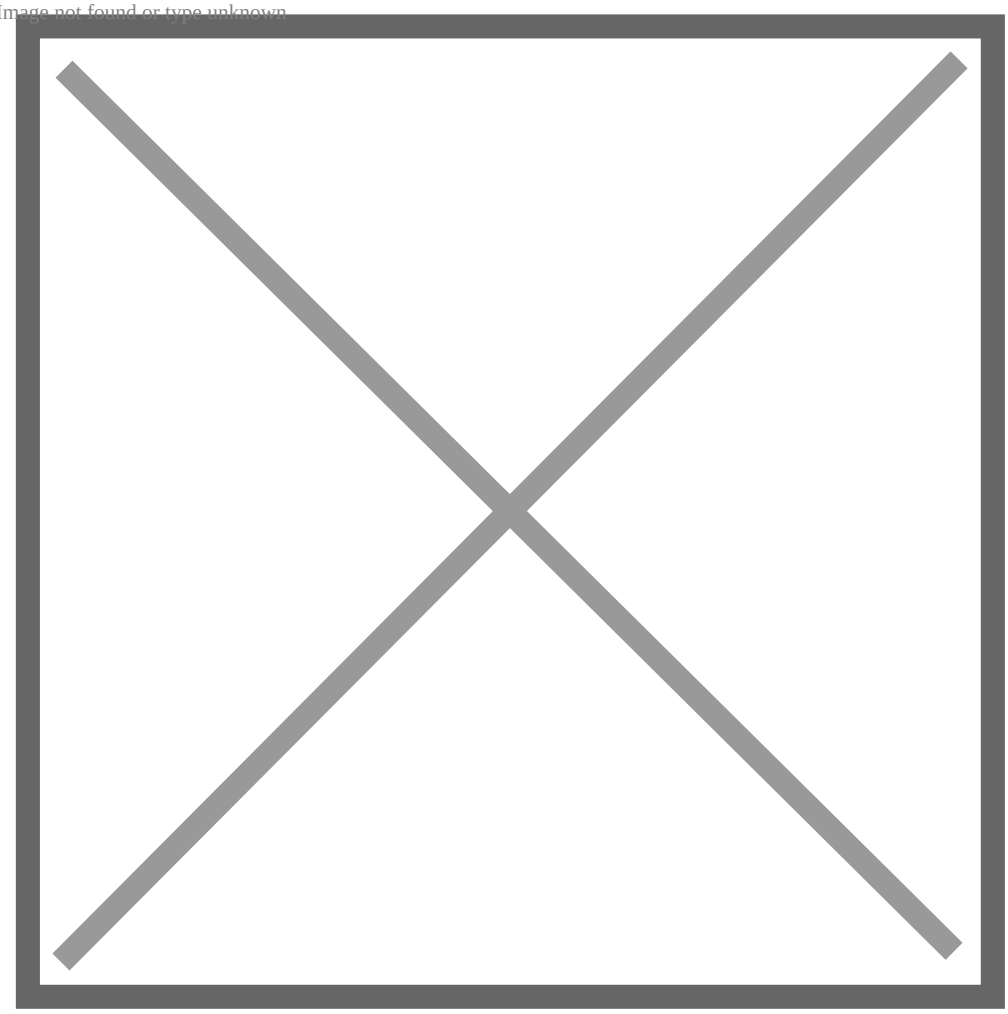
Image not found or type unknown



Natija: ['Anvar', 'Bexruz', 'Karim', 'Salim', 'Sobir', 'Tolib']

8. Ro'yxatni teskari tartibda tartiblash uchun .sort() metodi ichida reverse=True argumentini kiritamiz.

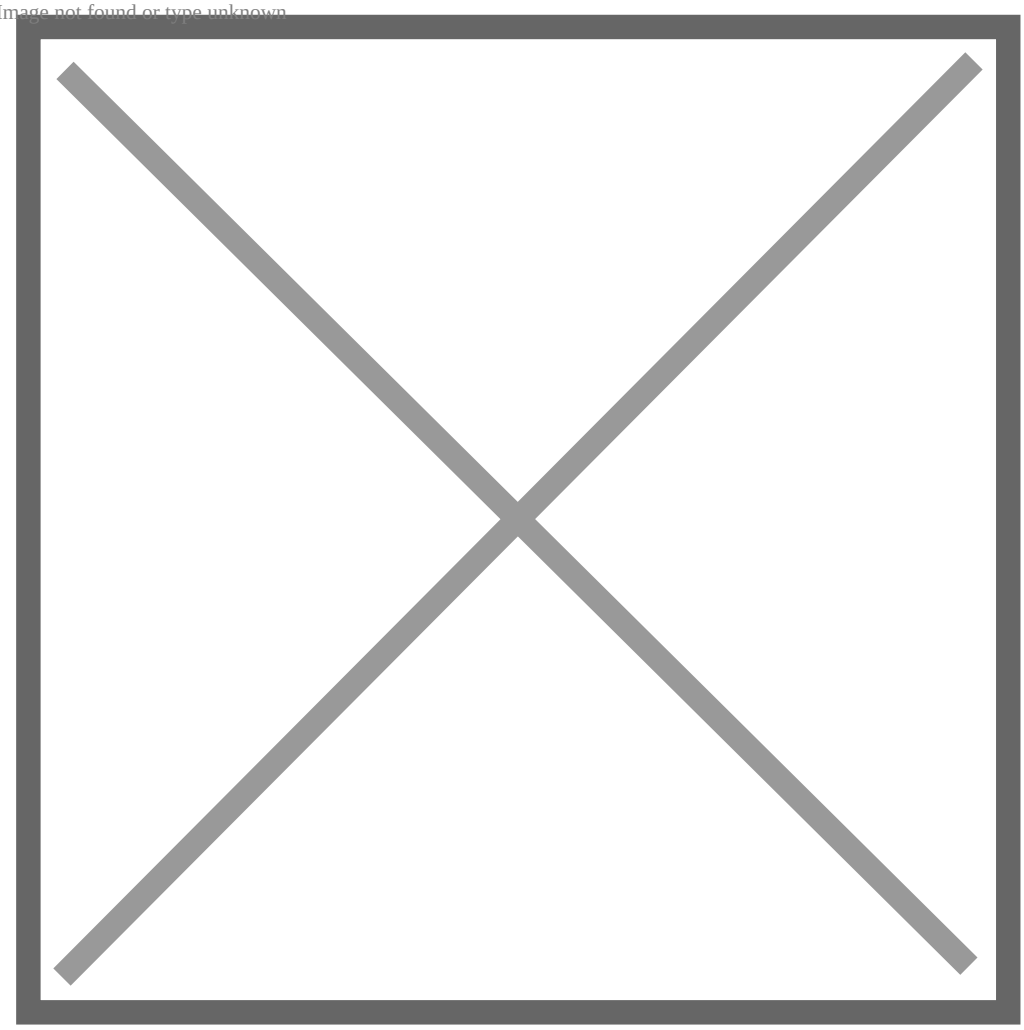
Image not found or type unknown



Natija: ['Tolib', 'Sobir', 'Salim', 'Karim', 'Bexruz', 'Anvar']

9. sorted() funksiyasi yordamida ham yuqoridagi amallarni bajarishimiz mumkin.

Image not found or type unknown



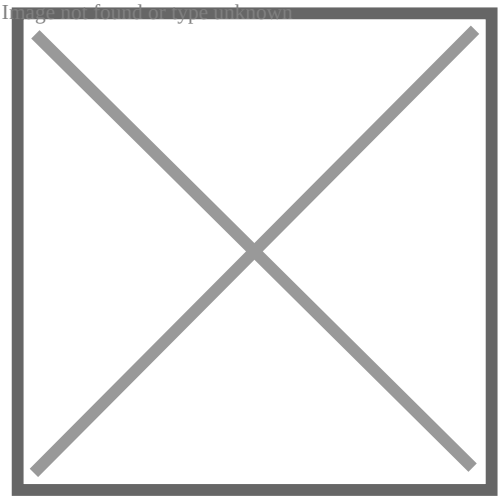
Natija:

['Anvar', 'Bexruz', 'Karim', 'Salim', 'Sobir', 'Tolib']

['Tolib', 'Sobir', 'Salim', 'Karim', 'Bexruz', 'Anvar']

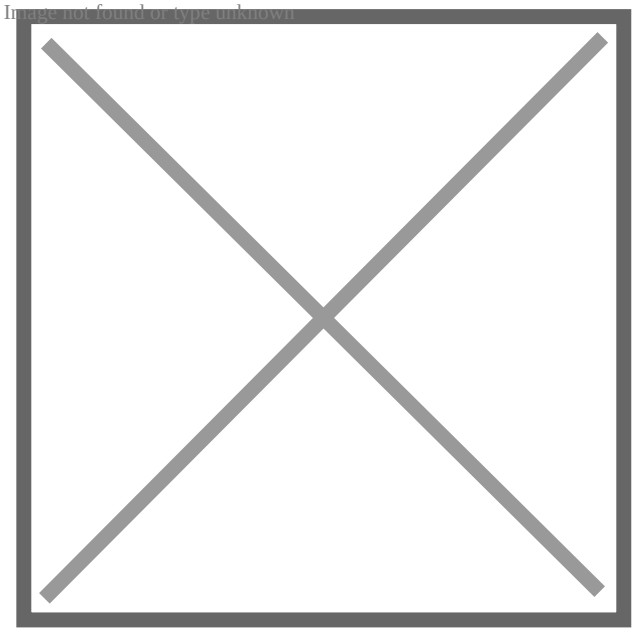
10.reverse() metodi yordamida ro’yxatni elentlarini teskarisiga aylantirishda foydalanamiz.

Image not found or type unknown



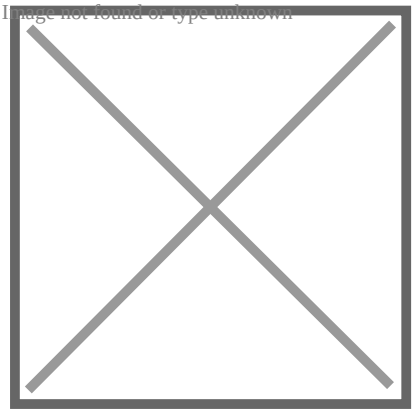
Natija: [93, 6, 12, 54, 78, 45]

11. Ro’yxat elemetlar sonini aniqlash uchun len() funksiyasidan fordalanamiz.



Natija: Elementlar soni: 6

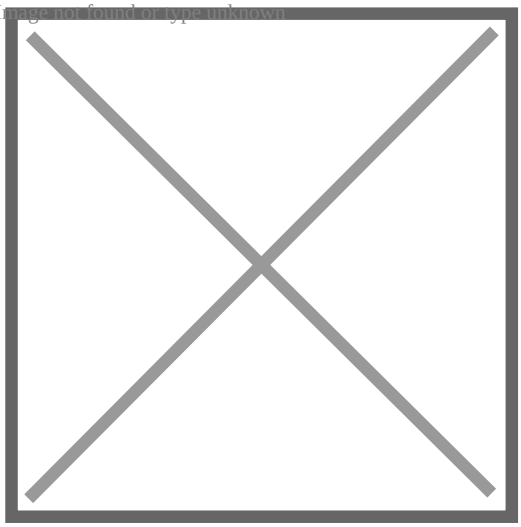
12. range() funktsiya yordamida biz ma'lum oraliqdagi sonlar ketma-ketligini yaratishimiz mumkin. **list()** funktsiyasi yordamida esa bu oraliqni ro'yxat shaklida saqlab olamiz:



Nataja: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

13. range() funksiyasida qadamni ham ko’rsatishimiz mumkin.

Image not found or type unknown



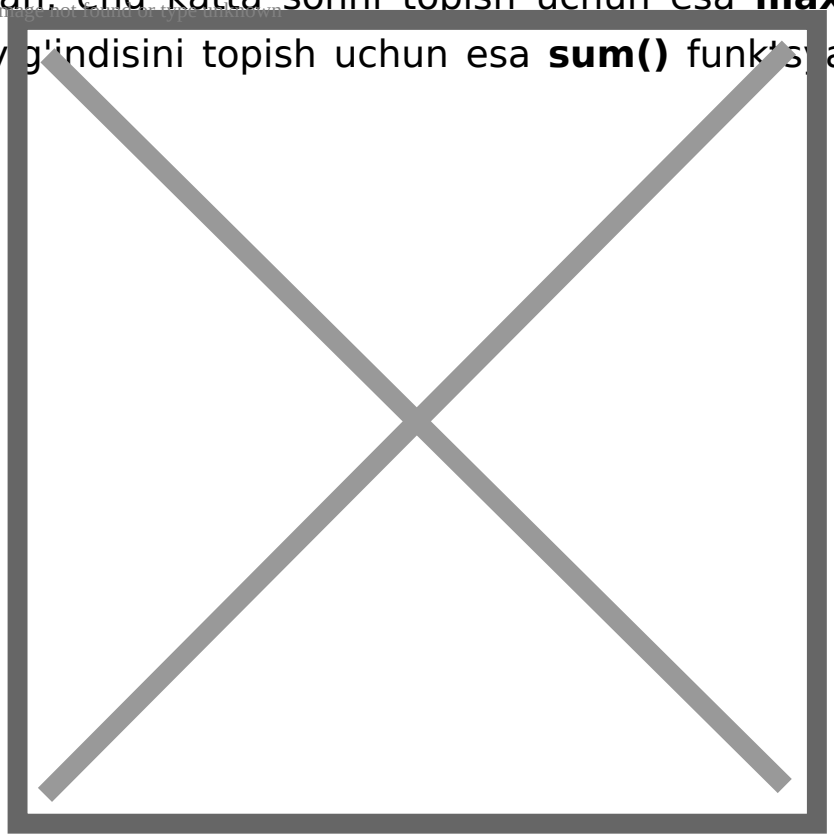
Natija:

[0, 2, 4, 6, 8, 10, 12, 14]

[1, 3, 5, 7, 9, 11, 13]

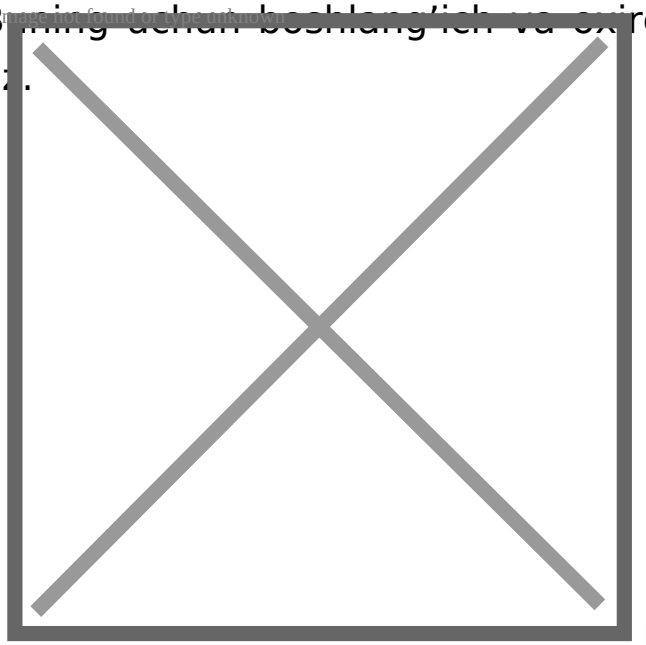
14. Pythonda ro'yxatdagi eng kichik sonni topish uchun **min()** funktsiyasidan, eng katta sonni topish uchun esa **max()** funktsiyasidan, sonlarning yig'indisini topish uchun esa **sum()** funktsiyasidan foydalanish mumkin:

Image not found or type unknown



Natija:

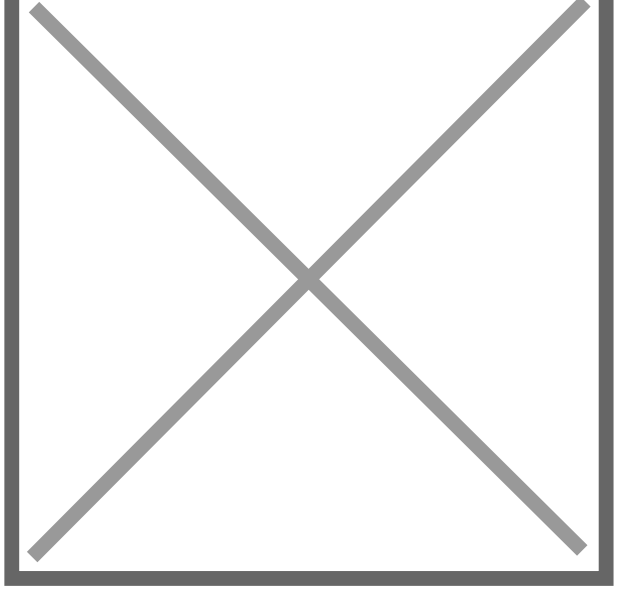
15. Ba'zida ro'yxatning ma'lum bir bo'lagini ajratib olish talab qilinishi mumkin, masalan sonlar ro’yxatidan 4 ta elemetdan yangi ro’yxat tuzamiz. Buning uchun boshlang’ich va oxirgi elemtlar oralig’ini indeksini ko’rsatamiz.



Natija: [7, 9, 74, 85]

TUPLES - O'ZGARMAS RO'YXAT

Dastur yaratish davomida o'zgarmas ro'yxat tuzish talab qilinishi mumkin. Pythonda bunday ro'yxatlar **tuples** deb yuritiladi. Tuple ichidagi qiymatlarni bir marta, dastur boshida beriladi va so'ngra o'zgartirib bo'lmaydi. List dan farqli ravishda, Tuple e'lon qilishda kvadrat qavslar [] o'rniga oddiy qavslar () ishlatiladi. Tuple ichidagi elementlarga huddi ro'yxat elementlariga murojat qilingani kabi murojat qilinaveradi:



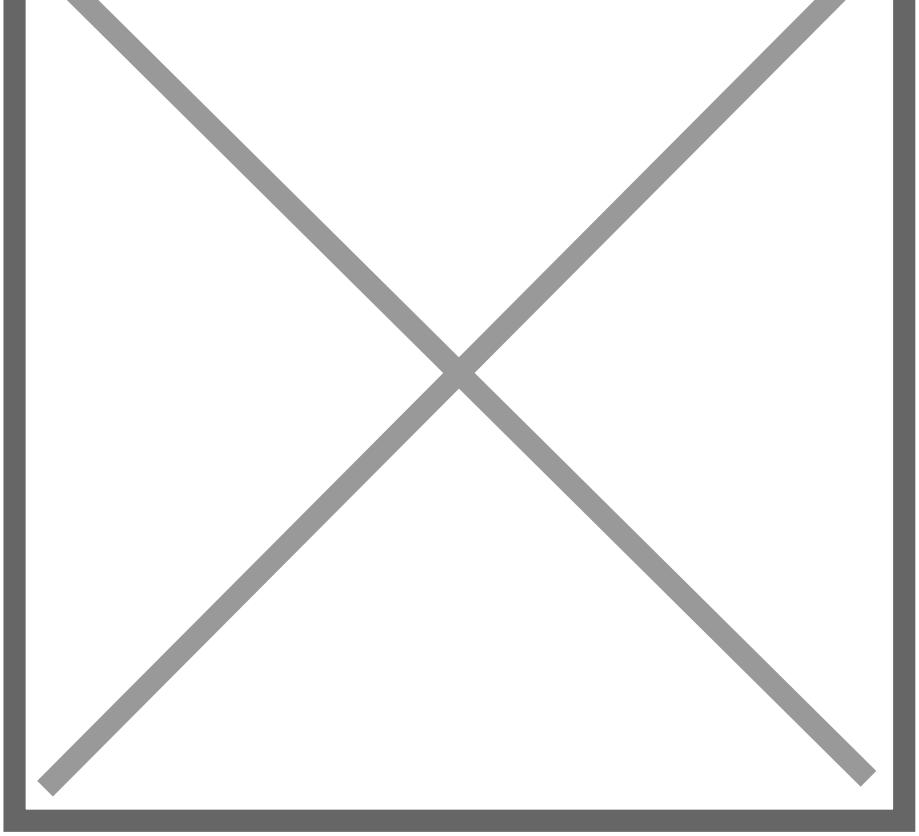
Natija:

(45, 12, 7, 9, 74, 85, 97, 67)

7

85

Agar Tuple ga o'zgartirish talab qilinsa, yagona yo'li o'zgarmas ro'yxatni list() funktsiyasi yordamida **List** (oddiy ro'yxat) ko'rinishiga keltirib olish, o'zgarishlarni bajarsih va qaytarib tuple() funktsiyasi yordamida o'zgarmas ro'yxatga o'tkazish mumkin:



Natija:

[45, 12, 7, 9, 74, 85, 97, 67, 89]

(45, 12, 7, 9, 74, 85, 97, 67, 89)

MUSTAQIL BAJARISH UCHUN TOPSHIRIQLAR

Masala 1. A=[123, 2, 65, 98, 13] va B=[76, 21, 67, 95] ro’yxatni tashkil etish dasturi tuzilsin.

Masala 2. A=[123, 2, 65, 98, 13] ro’yxatiga 29, 53 sonlarini qo’shish dasturi tuzilsin.

Masala 3. A=[123, 2, 65, 98, 13] ro’yxatida A[1] va A[4] elementlarini qo’shish dasturi tuzilsin.

Masala 4. List nomli ro’yxat tashkil eting va ro’yxat oxirida element qo’shish dasturi tuzilsin.

Masala 5. Mevalar nomlaridan iborat ro’yxat tashkil etish dasturi tuzilsin.

Masala 6. Meva=[‘anor’, ‘nok’, ‘olma’] ro’yxatiga ‘olcha’ elementini qo’shish dasturi tuzilsin.

Masala 7. Meva=[‘anor’, ‘nok’, ‘olma’] ro’yxatidan birinchi elementni o’chirish dasturi tuzilsin.

Masala 8. Meva=[‘anor’, ‘nok’, ‘olcha’, ‘olma’] ro’yxatidan .pop() metodi yordamidan foydalanish dasturi tuzilsin.

Masala 9. Harflar=[‘f’, ‘v’, ‘d’, ‘e’, ‘z’, ‘p’] ro’yxati elementlarini alifbo tartibida chiqarish dasturi tuzilsin.

Masala 10. Harflar=[‘f’, ‘v’, ‘d’, ‘e’, ‘z’, ‘p’] ro’yxati elementlarini teskari tartibida chiqarish dasturi tuzilsin.

Masala 11. Harflar=[‘f’, ‘v’, ‘d’, ‘e’, ‘z’, ‘p’] ro’yxati elementlarini .sorted() metodidan foydalanib teskari tartibida chiqarish dasturi tuzilsin.

Masala 12. Meva=[‘anor’, ‘nok’, ‘olcha’, ‘olma’] ro’yxati elementlarini alifbo tartibida chiqarish dasturi tuzilsin.

Masala 13. Harflar=[‘f’, ‘v’, ‘d’, ‘e’, ‘z’, ‘p’] ro’yxati elementlari sonini aniqlash dasturi tuzilsin.

Masala 14. (2, 25) oraliqdan bo’lgan ro’yxatni tashkil etish dasturi tuzilsin.

Masala 15. (2, 25) oraliqdan 3 qadam bilan ro’yxatni bilan tashkil etish dasturi tuzilsin.

Masala 16. (13, 34) oraliqdan 5 qadam bilan ro’yxatni bilan tashkil etish dasturi tuzilsin.

Masala 17. Sonlar=[23, 4, 56, 12, 86, 231] ro’yxatini elementlaridan eng kattasini topish dasturi tuzilsin.

Masala 18. Sonlar=[23, 4, 56, 12, 86, 231] ro’yxatini elementlaridan eng kichigini topish dasturi tuzilsin.

Masala 19. Sonlar=[23, 4, 56, 12, 86, 231] ro’yxatini elementlarini yig’indisini topish dasturi tuzilsin.

Masala 20. Sonlar=[23, 4, 56, 12, 86, 231] ro’yxatini [1:3] elementlarini ajratib olib yangi ro’yxat tashkil etish dasturi.

Masala 21. A=(123, 2, 65, 98, 13) o’zgarmas ro’yxatni tashkil etish dasturi tuzilsin.

Masala 22. A=[123, 2, 65, 98, 13] ro’yxatini o’zgarmas ro’yxatga aylantirish dasturini tuzilsin.

Masala 23. B=[76, 21, 67, 95] ro’yxatini o’zgarmas ro’yxatga aylantirish dasturini tuzilsin.

Masala 24. A=(123, 2, 65, 98, 13) o’zgarmas ro’yxatini ro’yxatga aylantirish dasturini tuzilsin.

Masala 25. B=(76, 21, 67, 95) o’zgarmas ro’yxatini ro’yxatga aylantirish dasturini tuzilsin.