

1. $S = 0,5 + 1,5 + 2,5 + \dots + 98,5 + 99,5$ ifodani hisoblash dasturini tuzing.
2. n natural sonining barcha bo'luvchilarini chiqaruvchi dastur tuzing.
3. Kiritilgan n soni qancha raqamdan iborat ekanligini aniqlovchi dastur tuzing (ko'rsatma: $n = n\%10$ ifoda $n = 0$ bo'lguncha bajariladi).
4. Kiritilgan n soni raqamlari yig'indisini hisoblovchi dastur tuzing.
5. Kiritilgan n soni juft raqamlari sonini hisoblovchi dastur tuzing.
6. n natural soni berilgan. 1 dan n gacha bo'lgan natural sonlar ichida oxirgi raqami 3 ga karrali sonlarni chiqaruvchi dastur tuzing.

55-DARS. SIKLLARNI BOSHQARISH: CONTINUE, BREAK OPERATORLARI

Sikllarni boshqarish uchun **break** va **continue** kabi maxsus operatorlardan foydalaniladi. **break** operatori sikldan chiqish uchun ishlatiladi. **continue** operatori siklning navbatdagi iteratsiyasiga o'tish uchun ishlatiladi.

1. Cheksiz siklni to'xtatishning Ctrl+C dan boshqa usuli mavjudmi?
2. Sikl to'xtaganidan keyin uni davom ettirish mumkinmi?

Sikldan chiqish

Agar shart True qiymat qaytarsa ham, sikl ichida **break** operatoriga murojaat etilsa, u holda sikl ishi to'xtatiladi. Sikl tarkibidagi ixtiyoriy buyruq **break**ga murojaat etilganidan keyin bekor qilinadi.

Jayanch tushunchalar

break – sikl ishini to'xtatuvchi operator.
continue – joriy siklni o'tkazib yuborib, keyingisiga o'tuvchi operator.

Misol. Karra jadvali bo'yicha o'quvchilar bilimini tekshiramiz.

```
table=8
for i in range(1,11):
    print(table, 'x', i, '= ?' )
    pup=input()
    res=table*i
    if int(pup)==res:
        print('Barakalla!')
    else:
        print('Noto\'gri, javob:', res)
print('Tugadi')
```

```
8 x 1 = ?
5
Noto'gri, javob: 8
```

i 1 dan 11 gacha bo'lgan 10 ta siklni hisoblaydi. Karra jadvali savolini chiqaradi. O'quvchi kiritgan javobni qabul qiladi. Ko'paytmani hisoblaydi. Natijani o'quvchi kiritgan javob bilan taqqoslaydi. To'g'ri bo'lsa, **Barakalla** yozuvini chiqaradi. Aks holda, noto'g'riligi haqida xabar berib, tog'ri javobni chiqaradi.

Sikldan chiqish uchun dastur kodiga **break** operatorini qo'shamiz. O'quvchi **bilmayman** so'zini yozgan vaqtda, sikl ishini to'xtatadi.

```

table=8
for i in range(1,11):
    print(table, 'x', i, '= ?' )
    pup=input()
    if pup=='bilmayman':
        break
    res=table*i
    if int(pup)==res:
        print('Barakalla!')
    else:
        print('Noto\'gri, javob:', res)
print('Tugadi')

```

```

8 x 1 = ?
8
Barakalla!
8 x 2 = ?
16
Barakalla!
8 x 3 = ?
bilmayman
Tugadi
>>>

```

Agar o'quvchi **bilmayman** deb javob bersa, dastur sikldan chiqadi va **Tugadi** yozuvini chiqaradi.

Siklni davom ettirish

continue operatori yordamida, sikldan chiqmagan holda, savolni o'tkazib yuborib, jarayonni davom ettirish mumkin. Agar ushbu operator sikl o'rtasida uchrasa, sikl oxirigacha bo'lgan keyingi buyruqlarni o'tkazib yuboradi va sikl o'z ishini keyingi iteratsiyadan davom ettiradi.

```

table=8
for i in range(1,11):
    print(table, 'x', i, '= ?' )
    pup=input()
    if pup=='bilmayman':
        break
    if pup=='keyingisi':
        print('Keyingi savol')
        continue
    res=table*i
    if int(pup)==res:
        print('Barakalla!')
    else:
        print('Noto\'gri, javob:', res)
print('Tugadi')

```

Agar o'quvchi **"keyingisi"** javobini kiritrsa, dastur siklni keyingisiga o'tkazib yuboradi.

```

8 x 1 = ?
8
Barakalla!
8 x 2 = ?
16
Barakalla!
8 x 3 = ?
keyingisi
Keyingi savol
8 x 4 = ?
32
Barakalla!
8 x 5 = ?

```

O'zgaruvchilarning o'zaro qiymat almashtirishi

Python dasturlash tilida o'zgaruvchilar qiymatini eng kam qadamlar bilan osongina o'zgartirish mumkin.

Masalan,

```
>>> a, b = 0, 1
```

```

a = 0
b = 1

```

Odatda, bu amal ikkita o'zgaruvchi qiymatini bir vaqtda o'zgartirish uchun kerak bo'ladi.