

## Зертханалық жұмыс №6 Цикл for және while

**Мамандық:** Деректер туралы ғылым

**Пән атауы:** Алгоритмдер және деректер құрылымы

**Оқытушы:** Базарбек Жания Пархатқызы

Python-да циклдердің екі негізгі түрі бар:

- белгілі бір рет қайталанатын циклдар (үшін, санау циклдері);
- белгілі бір оқиға болғанша қайталанатын циклдар (әзірше, шартты циклдар).

Егер біз қанша итерация жасау керектігін алдын ала білсек, for циклі тамаша жұмыс істейді.

«Сәлем» сөзін 10 рет басып шығаратын кодты қарастырыңыз:

```
for i in range(10):  
    print('Сәлем')
```

for операторының жолының соңындағы қос нүкте (:) Python интерпретаторына келесі командалар блогы бар екенін айтады. Пәрмен блогы for операторы бар жолдан шегініссіз келесі жолға дейінгі шегіністердің барлық жолдарын қамтиды.

Алдыңғы сабақтарда біз бірнеше енгізу пәрмендері арқылы бірнеше сандарды оқыдық. For циклін қолдану арқылы сіз қалағаныңызша санды оқып, өңдей аласыз.

Келесі бағдарлама кодын қарастырыңыз:

```
for i in range(5):  
    num = int(input())  
    print('Санның квадраты:', num * num)  
print('Цикл аяқталды')
```

Мұндай бағдарлама 5 санды оқиды және олардың квадраттарын түсіндірме мәтінмен бірге экранда көрсетеді. Екінші және үшінші жолдар шегініс болғандықтан, Python мұны 5 рет орындалатын циклдің денесі деп санайды. Төртінші жол шегініссіз, сондықтан ол циклдің бөлігі емес және цикл аяқталғаннан кейін тек бір рет орындалады.

### for циклін қолдану мысалдары

Келесі бағдарлама кодын қарастырыңыз:

```
print('A')  
print('B')  
for i in range(5):  
    print('C')  
    print('D')  
print('E')
```

Мұндай бағдарламаны орындаудың нәтижесі сызықтар болады

A  
B  
C  
D  
C

D  
C  
D  
C  
D  
C  
D  
E

Мысалы:

```
print('A')  
print('B')  
for i in range(5):  
    print('C')  
for i in range(5):  
    print('D')  
print('E')
```

Мұндай бағдарламаны орындаудың нәтижесі келесі жолдар болады:

A  
B  
C  
C  
C  
C  
D  
D  
D  
D  
D  
E

### Функция range()

<code>range(10)</code>	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
<code>range(1, 10)</code>	1, 2, 3, 4, 5, 6, 7, 8, 9
<code>range(3, 7)</code>	3, 4, 5, 6
<code>range(7, 3)</code>	пустая последовательность
<code>range(2, 15, 3)</code>	2, 5, 8, 11, 14
<code>range(9, 2, -1)</code>	9, 8, 7, 6, 5, 4, 3
<code>range(3, 10, -2)</code>	пустая последовательность

Мысалы:

```
for i in range(5, 0, -1):
    print(i, end=' ')
print('Бум!!!')
```

Нәтиже:

```
5 4 3 2 1 Бум!!!
```

Көбінесе бір нәрсенің қанша рет болғанын санау үшін біздің бағдарламаларымыз қажет. Мысалы, бейне ойын кейіпкердің неше рет айналатынын санауы мүмкін немесе математикалық бағдарлама қанша санның белгілі бір қасиеті бар екенін санауы мүмкін. Санау кілті санауыш айнымалыны пайдалану болып табылады.

10 санын оқып, олардың қаншасы 10-нан үлкен екенін анықтайтын программа жазайық.

```
counter = 0
for _ in range(10):
    num = int(input())
    if num > 10:
        counter = counter + 1
print('енгізілген, counter, '10 нан үлкен')
```

Есептегіш айнымалыны құру және оған бастапқы мән беру: `counter = 0`;

Есептегіш айнымалыны 1-ге арттыру: `counter = counter + 1`.

Тағы бір мысалды қарастырайық: [1;100] аралықтағы квадраты 4-ке аяқталатын сандар санын есептеңіз.

```
counter = 0
for i in range(1, 101):
    if i**2 % 10 == 4:
        counter = counter + 1
print(counter)
```

Айнымалы мәндердің алмасуы

```
x, y = y, x
```

Осы кодты орындау нәтижесінде Python x және y айнымалыларының мәндерін ауыстырады.

Натурал санның жай сан екенін анықтайтын программа жазайық:

```
num = int(input())
flag = True

for i in range(2, num):
    if num % i == 0:
        flag = False

if num == 1:
    print('Бұл бірлік, ол жай да, күрделі де емес')
elif flag == True:
    print('жай сан')
```

```
else:
    print('Құрама сан')
```

### Әртүрлі тапсырма мәлімдемелері x=6

Инструкция	Что она делает	Значение x после инструкции
<code>x = x + 4</code>	Прибавляет 4 к <code>x</code>	10
<code>x = x - 3</code>	Вычитает 3 из <code>x</code>	3
<code>x = x * 10</code>	Умножает <code>x</code> на 10	60
<code>x = x / 4</code>	Делит <code>x</code> на 4	1.5
<code>x = x // 4</code>	Делит нацело <code>x</code> на 4	1
<code>x = x % 4</code>	Находит остаток от деления <code>x</code> на 4	2

Оператор	Пример использования	Эквивалент
<code>+=</code>	<code>x += 5</code>	<code>x = x + 5</code>
<code>--</code>	<code>x -= 2</code>	<code>x = x - 2</code>
<code>*=</code>	<code>x *= 10</code>	<code>x = x * 10</code>
<code>/=</code>	<code>x /= 4</code>	<code>x = x / 4</code>
<code>//=</code>	<code>x //= 4</code>	<code>x = x // 4</code>
<code>%=</code>	<code>x %= 4</code>	<code>x = x % 4</code>

### Тапсырма

1.  $K$  және  $N$  ( $N > 0$ ) бүтін сандары берілген.  $N$  санын  $K$  санын көбейтіңіз.
2. Екі бүтін  $A$  және  $B$  ( $A < B$ ) саны берілген.  $A$  және  $B$  арасында орналасқан барлық бүтін сандарды (соның ішінде  $A$  және  $B$  сандарының өзін), сондай-ақ осы сандардың  $N$  санын өсу ретімен шығарыңыз.
3. Екі бүтін  $A$  және  $B$  ( $A < B$ ) саны берілген.  $A$  және  $B$  арасындағы барлық бүтін сандарды ( $A$  және  $B$  сандарын қоспағанда), сондай-ақ осы сандардың  $N$  санын кему ретімен басып шығарыңыз.

4. Нақты санды ескере отырып - 1 кг тәттілердің бағасы. Шығару құны 1, 2, . . . , 10 кг тәттілер.
5. Екі бүтін  $A$  және  $B$  ( $A < B$ ) саны берілген.  $A$ -дан  $B$ -ға дейінгі барлық бүтін сандардың қосындысын табыңыз.
6. Екі бүтін  $A$  және  $B$  ( $A < B$ ) саны берілген.  $A$ -дан  $B$ -ға дейінгі барлық бүтін сандардың көбейтіндісін табыңыз.
7. Екі бүтін  $A$  және  $B$  ( $A < B$ ) саны берілген.  $A$ -дан  $B$ -ға дейінгі барлық бүтін сандардың квадраттарының қосындысын табыңыз.
8.  $N$  ( $> 0$ ) бүтін саны берілген.  $1 + 1/2 + 1/3 + \dots + 1/N$  қосындысын табыңыз.
9.  $N$  ( $> 0$ ) бүтін саны берілген. 1.1 1.2 1.3 көбейтіндісін тап. . . ( $N$  фактор).
10.  $N$  және  $K$  натурал сандар берілген.  $1K + 2K + \dots + NK$  қосындысын табыңыз. Бүтін сандардың толып кетуін болдырмау үшін нақты айнымалыны пайдаланып осы қосындының шарттарын есептеңіз және нәтижені нақты сан ретінде шығарыңыз.
11.  $N$  ( $> 0$ ) бүтін саны берілген.  $11 + 22 + \dots + NN$  қосындысын табыңыз. Бүтін сандардың толып кетуін болдырмау үшін нақты айнымалыны пайдаланып осы қосындының шарттарын есептеңіз және нәтижені нақты сан ретінде шығарыңыз.
12.  $N$  ( $> 0$ ) бүтін саны берілген.  $1N + 2N-1 + \dots + N1$  қосындысын табыңыз. Бүтін сандардың толып кетуін болдырмау үшін нақты айнымалыны пайдаланып осы қосындының шарттарын есептеңіз және нәтижені нақты сан ретінде шығарыңыз.
13.  $A$  және  $B$  ( $A < B$ ) натурал сандары берілген.  $A$ -дан  $B$ -ға дейінгі барлық бүтін сандарды қоса алғанда шығару; әрбір сан оның мәніне қанша рет болса, сонша көрсетілуі керек. (мысалы, 3 саны 3 рет көрсетіледі).
14.  $A$  және  $B$  ( $A < B$ ) бүтін сандары берілген.  $A$ -дан  $B$ -ға дейінгі барлық бүтін сандарды қоса алғанда шығару;  $A$  саны 1 рет көрсетілуі керек, ал  $A + 1$  саны 2 рет көрсетілуі керек, т.б.

### Цикл while

Біз санау циклі болып табылатын for циклінің жұмысын қарастырдық. Егер біз қанша итерация жасау керектігін алдын ала білсек, for циклі тамаша жұмыс істейді. Бірақ кейде цикл белгілі бір оқиға болғанша орындалуы керек және бұл жағдайда итерациялар санын алдын ала бағалау мүмкін емес. Бұл жерде уақытша цикл көмекке келеді.

Сәлем сөзін 10 рет басып шығаратын while циклін қолданатын кейбір кодты қарастырайық:

```
i = 0
while i < 10:
    print('Сәлем')
    i += 1
```

For және while циклінің көмегімен 3-ке еселік барлық сандарды басып шығаратын программа жазайық:

*For*

```
for i in range(0, 100, 3):
    print(i)
```

*while*

```
i = 0
while i < 100:
    print(i)
    i += 3
```

Пайдаланушы stop сөзін енгізгенше сандарды оқитын және олардың қосындысын табатын программа жазайық:

```
text = input()
total = 0
while text != 'stop':
    num = int(text)
    total += num
    text = input()
print(total)
```

Шексіз циклдің мысалы:

```
i = 0
total = 0
while i < 10:
    total += i
```

i айнымалысы цикл денесінде өзгермейтіндіктен,  $i < 10$  шарты ақиқат болып қалады және цикл шексіз көп рет орындалады.

Шексіз циклдар break операторымен бірге қолданылуы мүмкін.

## Санның цифрларын өңдеу

Бүтін сандарды (int деректер түрі) зерттегенде біз бүтін санды бөлу операциясы // және бір бүтін санды екінші % бөлудің қалдығын табу операциясы туралы айттық. Уақыт циклін және екі деректер операциясын пайдалана отырып, цифрлардың (цифрлардың) ерікті саны бар санның цифрларын өңдеуге болады.

Натурал n саны берілсін. Содан кейін:

- операцияның нәтижесі  $n \% 10$  – санның соңғы цифры;
- $n // 10$  операциясының нәтижесі соңғы цифрды жойылған сан болып табылады.

Натурал санды (оң бүтін) оқитын және оның цифрларын өңдейтін программа жазайық.

```
n = int(input())
while n != 0:
    last_digit = n % 10
    n = n // 10
```

while циклі санда өңделмеген сандар болғанша жұмыс істейді. Циклдің денесі мыналарды қамтиды:

- соңғы санды алу тәртібі  $соңғы\_сан = n \% 10$ ;
- соңғы санды өңдеу коды;
- $n = n // 10$  санынан соңғы цифрды алып тастау процедурасы.

Өңдеу процедурасы кез келген болуы мүмкін: сандарды шығару, қосындыны, сандардың көбейтіндісін табу, ең үлкен немесе ең кіші санды табу, қандай да бір шартты қанағаттандыратын сандарды санау және т.б.

Санда 7 саны бар-жоғын анықтайтын программа жазайық.

```
num = int(input())
has_seven = False

while num != 0:
    last_digit = num % 10
    if last_digit == 7:
        has_seven = True
    num = num // 10

if has_seven == True:
    print('YES')
else:
    print('NO')
```

### Тапсырма

1. Натурал сан берілген. Оның сандарын бағанға кері ретпен көрсететін программа жазыңыз.
2. Натурал сан берілген. Санның цифрларының орналасу ретін өзгертетін программа жазыңыз.
3.  $n$ , ( $n \geq 10$ ) натурал саны берілген. Оның ең үлкен және ең кіші сандарын анықтайтын программа жазыңыз.
4. Натурал сан берілген. Есептейтін программа жазыңыз:  
оның цифрларының қосындысы;  
ондағы цифрлардың саны;  
оның цифрларының көбейтіндісі;  
оның цифрларының арифметикалық ортасы;  
оның бірінші саны;  
оның бірінші және соңғы цифрларының қосындысы.
5.  $n$  ( $n > 9$ ) натурал саны берілген. Оның екінші (басынан) цифрын анықтайтын программа жазыңыз.

### Циклді үзу операторының үзілуі

Кейде циклды мерзімінен бұрын тоқтату қажет болуы мүмкін. break операторы ең жақын for немесе while циклін үзеді.

Break операторын қолданып, санның қарапайымдылығын тексеретін бағдарламаны жетілдіреміз:

```
num = int(input())
flag = True
for i in range(2, num):
    if num % i == 0:
        flag = False
        break
if flag:
```

```
print('Жай сан')
else:
    print('құрама сан')
```

10 санды оқитын және теріс санды тапқанша қосатын for циклін пайдаланып программа жазайық. Бұл жағдайда циклдің орындалуы break командасымен үзіледі:

```
result = 0
for i in range(10):
    num = int(input())
    if num < 0:
        break
    result += num
print(result)
```

Басқа стандартты циклдік идиома итерация кезінде жеке элементтерді өткізіп жіберу болып табылады. continue операторы цикл денесіндегі барлық пәрмендер аяқталғанға дейін for немесе while циклінің келесі итерациясына өтуге мүмкіндік береді.

7, 17, 29 және 78 сандарынан басқа 1-ден 100-ге дейінгі барлық сандарды басып шығаратын программа жазайық.

```
for i in range(1, 101):
    if i == 7 or i == 17 or i == 29 or i == 78:
        continue
    print(i)
```

## Тапсырма

1. Шаңғышы жаттығуды бірінші күні 10 шақырымға жүгіруден бастады. Әр келесі күні ол жүгіру ұзақтығын алдыңғы күннің жүгіруінің  $P$  пайызына арттырды ( $P$  - нақты,  $0 < P < 50$ ).  $P$  берілген, қай күннен кейін шаңғышының барлық күндердегі жалпы жүгірісі 200 км-ден асатынын анықтаңыз және табылған күндер санын  $K$  (бүтін) және жалпы жүгіріс  $S$  (нақты сан) шығарыңыз.
2. Бүтін сан берілген  $N (> 0)$ . Бөлу операцияларын қолдана отырып және бөлудің қалдықтарын алып, оның барлық цифрларын оң жақтан бастап шығарыңыз (бірліктер санаты).
3.  $N (> 0)$  бүтін саны берілген. Бүтін бөлу амалдарын қолданып, бөлудің қалған бөлігін алып, оның цифрларының санын және қосындысын табыңыз.
4.  $N (> 0)$  бүтін саны берілген. Бүтін бөлу амалдарын пайдаланып және бөлудің қалған бөлігін алып,  $N$  санын оңнан солға қарай оқу арқылы алынған санды табыңыз.
5.  $N (> 0)$  бүтін саны берілген. Бүтін санға бөлу амалдарын қолданып, бөлудің қалдығын алып,  $N$  санының жазбасында «2» санының бар-жоғын анықтаңыз. Бар болса, TRUE шығарыңыз, жоқ болса, ЖАЛҒАН шығарыңыз.
6.  $N (> 0)$  бүтін саны берілген. Бүтін бөлу амалдарын пайдаланып және бөлудің қалған бөлігін алып,  $N$  санының жазбасында тақ цифрлардың бар-жоғын анықтаңыз. Егер бар болса, ШЫНДЫҚ шығарыңыз, жоқ болса, ЖАЛҒАН шығарыңыз.
7.  $N (> 1)$  бүтін саны берілген. Егер ол жай болса, яғни 1 мен өзінен басқа оң бөлгіштері болмаса, онда TRUE деп басып шығарыңыз, әйтпесе ЖАЛҒАН деп басып шығарыңыз.



8. А және В бүтін оң сандары берілген. Евклид алгоритмін пайдаланып олардың ең үлкен ортақ бөлгішін (НОД) табыңыз:  $\text{НОД}(A, B) = \text{НОД}(B, A \bmod B)$ , если  $B \neq 0$ ;  $\text{НОД}(A, 0) = A$
9.  $N (> 1)$  бүтін саны берілген. Фибоначчи сандар тізбегі  $F_k$  келесідей анықталады:  $F_1 = 1, F_2 = 1, F_k = F_{k-2} + F_{k-1}, k = 3, 4, \dots$ .  $N$  Фибоначчи саны екенін тексеріңіз. Егер солай болса, онда TRUE шығарыңыз, егер жоқ болса, FALSE шығарыңыз.
10. Бүтін сан берілген  $N (> 1)$ . Бірінші Фибоначчи санын табыңыз,  $N$ -ден үлкен (Фибоначчи сандарының анықтамасы 29 тапсырмасында берілген).
11. Фибоначчи саны болып табылатын  $N (> 1)$  бүтін саны берілген:  $N = F_k$  (Фибоначчи сандарының анықтамасы 29 есепте берілген).  $F_{k-1}$  және  $F_{k+1}$  бүтін сандарын табыңыз — алдыңғы және келесі Фибоначчи сандары.
12. Фибоначчи саны болып табылатын  $N (> 1)$  бүтін саны берілген:  $N = F_k$  (Фибоначчи сандарының анықтамасы 29 есепте берілген).  $k$  бүтін санын табыңыз — Фибоначчи  $N$  санының реттік нөмірі.
13.  $\varepsilon$  нақты саны берілген ( $> 0$ ).  $A_k$  нақты сандар тізбегі келесідей анықталады:  $A_1 = 2, A_k = 2 + 1/A_{k-1}, k = 2, 3, \dots$ .  
 $|A_k - A_{k-1}| < \varepsilon$  шарты орындалатын бірінші  $k$  санын табыңыз, және осы санды, сондай-ақ  $A_{k-1}$  және  $A_k$  сандарын басып шығарыңыз.
14.  $\varepsilon$  нақты саны берілген ( $> 0$ ).  $A_k$  нақты сандар тізбегі келесідей анықталады:  $A_1 = 1, A_2 = 2, A_k = (A_{k-2} + 2 A_{k-1})/3, k = 3, 4, \dots$ .  
 $|A_k - A_{k-1}| < \varepsilon$  шарты орындалатын бірінші  $k$  санын табыңыз, және осы санды, сондай-ақ  $A_{k-1}$  және  $A_k$  сандарын басып шығарыңыз.
15. Берілген оң сандар  $A, B, C$ . Өлшемі  $A \times B$  тіктөртбұрышта  $C$  жағы бар мүмкіндігінше көп шаршы бар (қабатталмаған). Тіктөртбұрышқа орналастырылған шаршылардың санын табыңыз. Көбейту және бөлу амалдарын қолданбаңыз.