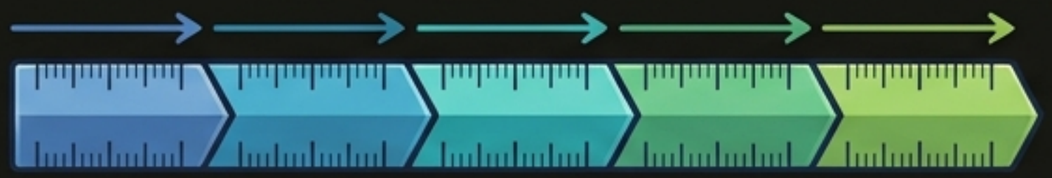
Python-дағы қайталану алгоритмдері: for және while циклдері

Практикалық есептер мен шешімдер (Интерактивті жұмыс дәптері)

Gemini Notebook

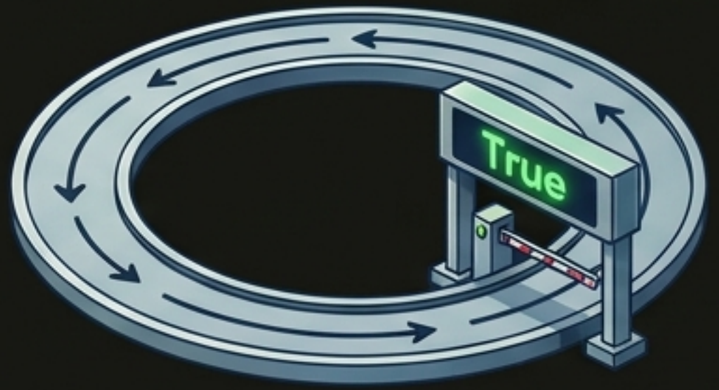
for циклі (Қайталау саны белгілі болғанда)



`range(1, 11)` – Сандар диапазонын құру.

```
for i in range(10):
```

while циклі (Қайталау шартқа байланысты болғанда)



Шарт ақиқат болғанша орындалады. Шарттың өзгеруін қамтамасыз ету маңызды (шексіз циклді болдырмау үшін).

```
while n > 0:
```

Gemini Notebook

break

Циклді толық тоқтатады.

```
if i == 6:
    break
```

```
if i == 6:
    break
```

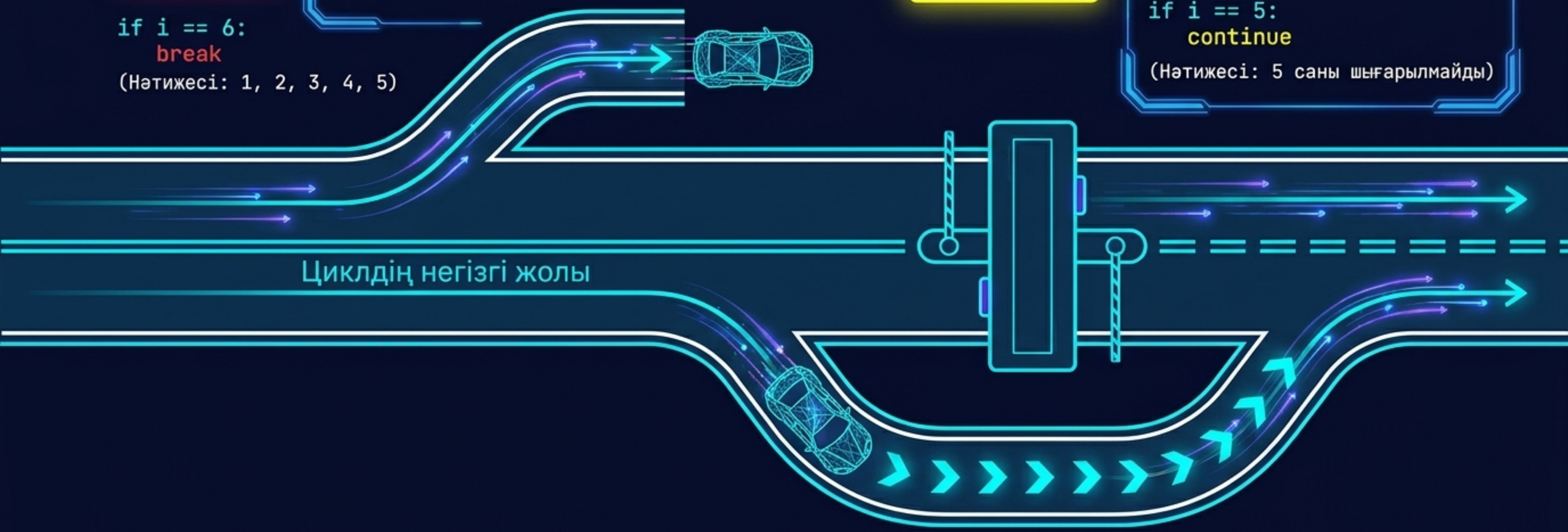
(Нәтижесі: 1, 2, 3, 4, 5)

continue

Ағымдағы қайталануды өткізіп, келесісіне көшеді.

```
if i == 5:
    continue
```

(Нәтижесі: 5 саны шығарылмайды)



Gemini Notebook

[Есеп 1] Санның цифрларының қосындысы

Пайдаланушы енгізген оң бүтін санның цифрларының қосындысын `while` циклі арқылы табыңыз.

```
> Сан енгізіңіз: 357
> Цифрлар қосындысы: 15 (өйткені 3 + 5 + 7 = 15)
```

⏸ **Келесі слайдқа өтпес бұрын өзіңіз шығарып көріңіз ...**

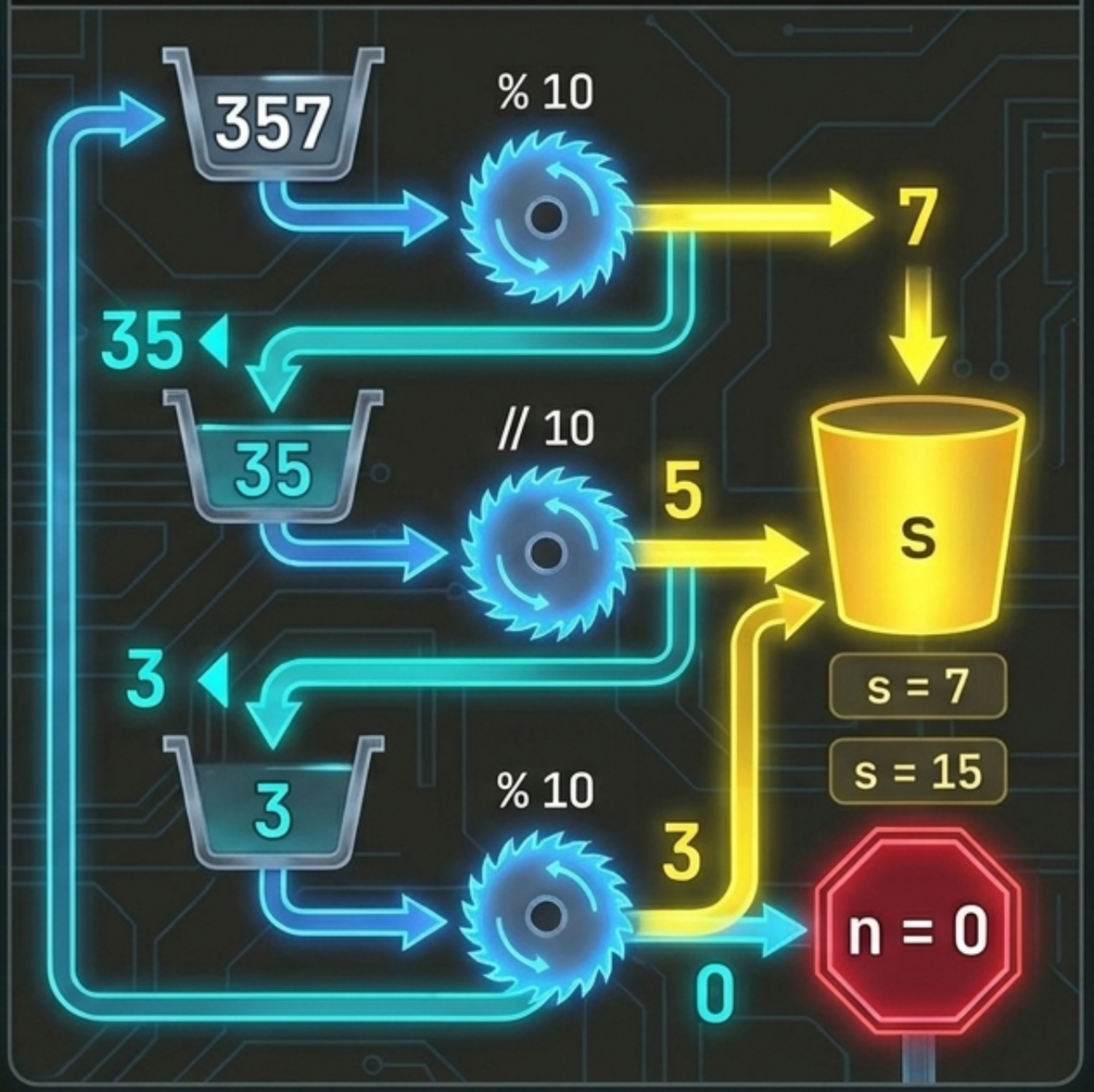
Gemini Notebook

Код өңдеушісі

```
n = int(input('Сан енгізіңіз: '))
s = 0
while n > 0:
    digit = n % 10
    s += digit
    n //= 10

print('Цифрлар қосындысы:', s)
```

Дебаггер визуализаторы



Gemini Notebook

[Есеп 2] Факториалды есептеу

Пайдаланушы енгізген n теріс емес бүтін санының факториалын `for` циклі арқылы есептеңіз.

```
> n = 5
> Факториал: 120
```

Ескерту: Факториал есептеу үшін жинақтаушы айнымалының бастапқы мәні 0 емес, 1 болуы керек.

⏸ **Кодты жазып көріңіз...**

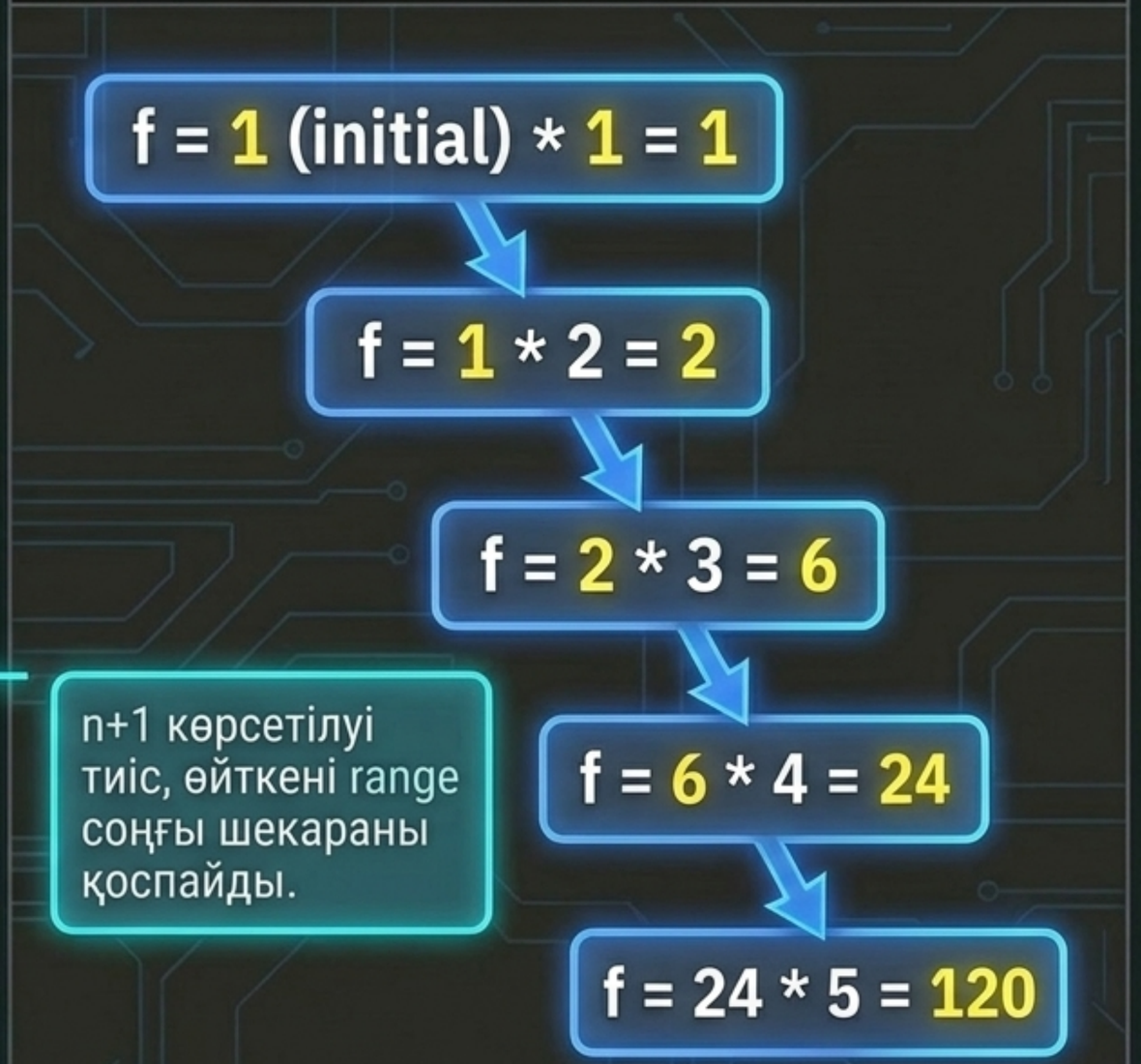
Gemini Notebook

Код өңдеушісі

```
1 n = int(input('n = '))
2 if n < 0:
3     print('Теріс сан енгізуге
4           болмайды')
5 else:
6     f = 1
7     for i in range(1, n + 1):
8         f *= i
9     print('Факториал:', f)
```

n+1 көрсетілуі тиіс, өйткені range соңғы шекараны қоспайды.

Дебаггер визуализаторы



Gemini Notebook

[Есеп 3] Жай санды тексеру

Пайдаланушы енгізген санның жай сан (тек 1-ге және өзіне ғана бөлінетін сан) екенін анықтаңыз.

Жай сандар: 2, 3, 5, 7, 11, 13. Жай емес сандар: 4, 6, 8, 9, 10.

```
> Сан енгізіңіз: 7
> Жай сан
> Сан енгізіңіз: 9
> Жай сан емес
```

⏸ **boolean жалаушасын (flag) қолданып көріңіз...**

Gemini Notebook

Код өңдеушісі

```
1 prime = True
2 for i in range(2, n):
3     if n % i == 0:
4         prime = False
5         break
6 if prime:
7     print('Жай сан')
8 else:
9     print('Жай сан емес')
```

Дебаггер визуализаторы



Gemini Notebook

[Есеп 4] Фибоначчи тізбегі

Пайдаланушы енгізген n санына сәйкес Фибоначчи тізбегінің алғашқы n мүшесін экранға шығарыңыз.
Негізгі формула: F(n) = F(n-1) + F(n-2).
Тізбек 0 және 1 сандарынан басталады.

```
> Мүшелер саны: 10
> 0 1 1 2 3 5 8 13 21 34
```

⏸ a және b айнымалыларын қалай жаңартамыз?

Gemini Notebook

Код өңдеушісі

```
1  n = int(input('Мүшелер саны: '))
2  a = 0
3  b = 1
4  for i in range(n):
5      print(a, end=' ')
6      a, b = b, a + b
7
```

Дебаггер визуализаторы

	Қадам	a мәні	b мәні	
1	Бастапқы	0	1	Print 0
2	Қадам 1	1	1	Print 1
3	Қадам 2	1	2	Print 1
4	Қадам 3	2	3	Print 2

Python-ның бір уақытта меншіктеу (simultaneous assignment) қасиетінің кербездігі.

Gemini Notebook

[Есеп 5] Сандарды санау (Шексіз цикл)

Пайдаланушы 0 санын енгізгенге дейін сандар енгізе береді. Енгізілген тек оң сандардың санын есептеңіз. Теріс сандар есепке алынбайды (өткізіп жіберіледі).

```
> 5
> -2 (өткізіледі)
> 10
> 0 (тоқтайды)
> Оң сандар саны: 2
```

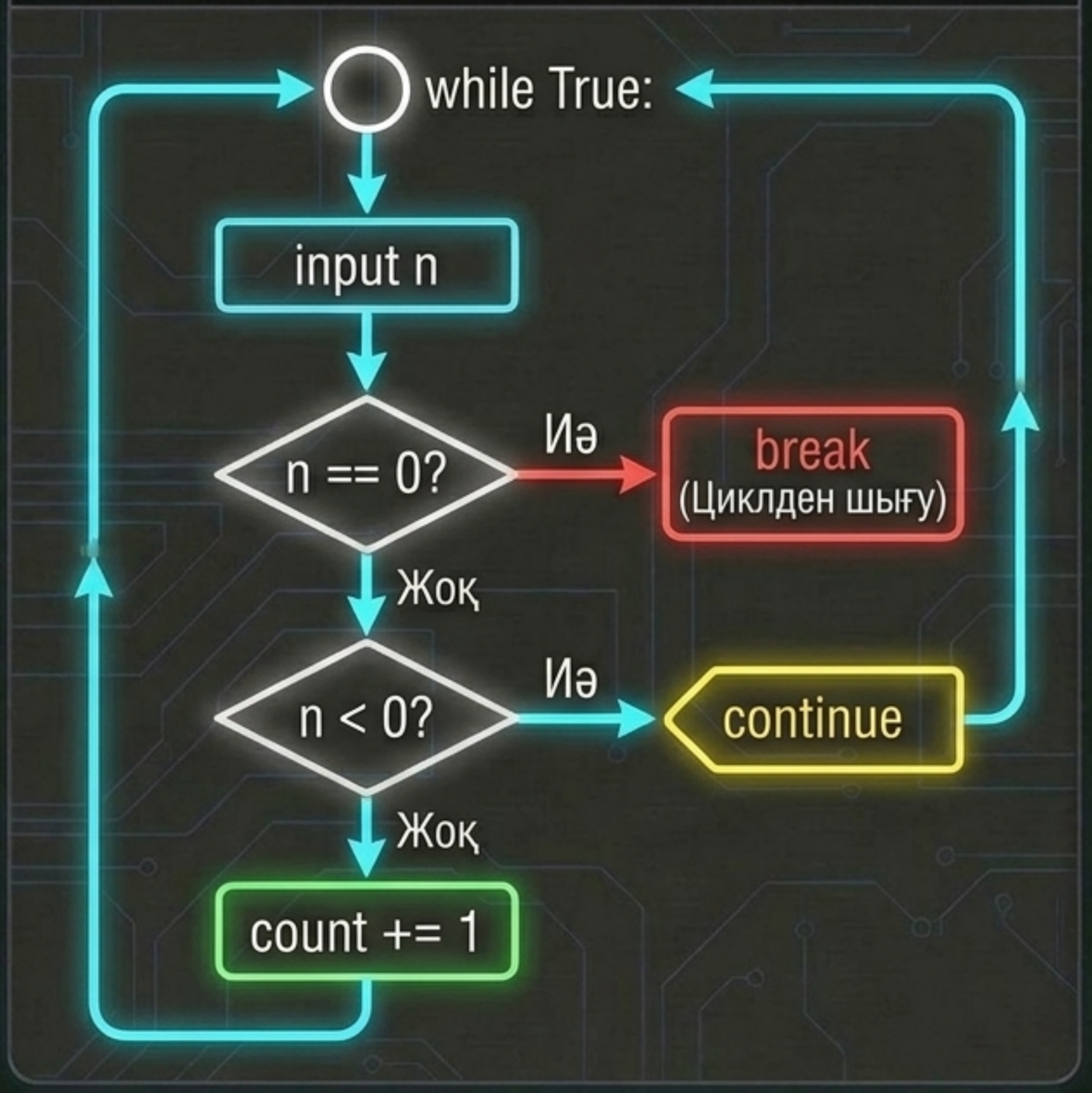
⏸ while True, break және continue біріктіріңіз...

Gemini Notebook

Код өңдеушісі

```
1  count = 0
2  while True:
3      n = int(input('Сан енгізіңіз: '))
4      if n == 0:
5          break
6      if n < 0:
7          continue
8      count += 1
9  print('Оң сандар саны:', count)
```

Дебаггер визуализаторы



Gemini Notebook

Циклдерді меңгерудің 3 негізгі қағидасы



Бастапқы және соңғы шарт

Қайталану саны мен тоқтау шартын нақты анықтаңыз. for циклінде range() шекараларын дұрыс көрсету маңызды.



Шексіз циклдің алдын алу

while циклін қолданғанда, цикл ішінде шартты өзгертетін (мысалы, i += 1, n //= 10) қадамның қадамның міндетті түрде болуын қадағалаңыз.



Дұрыс құралды таңдау

Массивтермен, белгілі диапазондармен жұмыс жасауда for циклін қолданыңыз. Оқиғаларға немесе күрделі шарттарға тәуелді есептер үшін while тиімді.

Gemini Notebook

```
>>> print('Циклдер – бағдарламалаудың жүрегі.')
```

Есептерді тек оқып қоймай, кодты өз қолыңызбен жазып, тәжірибе жасаңыз. 100 код жолын жазу 100 бет теория оқығаннан артық.

Gemini Notebook