



**ИНФОРМАТИКА**

**8**

класс

# ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ

НАЧАЛА ПРОГРАММИРОВАНИЯ  
НА ЯЗЫКЕ PYTHON

# КЛЮЧЕВЫЕ СЛОВА

- ◆ повторение
- ◆ циклический алгоритм
- ◆ оператор цикла
- ◆ тело цикла
- ◆ цикл с условием
- ◆ цикл с переменной
- ◆ вложенные циклы
- ◆ сложность алгоритма

# ПОВТОРЕНИЕ

**Повторение** (цикл) - последовательность действий, выполняемых многократно.

Последовательность действий, многократно повторяющаяся в процессе выполнения цикла, называется **телом цикла**.

Типы циклов:

1) цикл с условием:

- ◆ цикл с заданным условием продолжения работы;
- ◆ цикл с заданным условием окончания работы;

2) цикл с переменной.



**Циклический алгоритм** - алгоритм, содержащий последовательность действий, многократно повторяющуюся в процессе его выполнения.

**Оператор цикла** - команда, реализующая на языке программирования алгоритмическую конструкцию «повторение».

В языке программирования Python существует несколько операторов цикла:

◆ while

◆ for



# ПРОГРАММИРОВАНИЕ ЦИКЛОВ С ЗАДАНЫМ УСЛОВИЕМ ПРОДОЛЖЕНИЯ РАБОТЫ

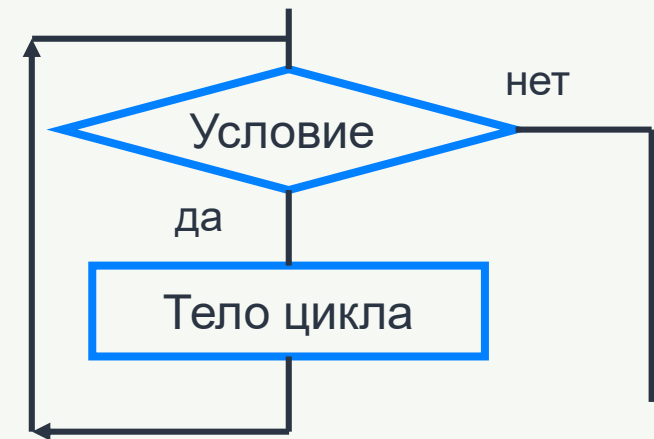
Общий вид оператора:

```
while <условие> :  
    <тело цикла>
```

Здесь:

<условие> - логическое выражение;  
пока оно истинно, выполняется тело цикла;

<тело цикла> - один или несколько операторов,  
описывающих последовательность действий,  
выполняемых многократно.



# ЧИТАЕМ ПРОГРАММНЫЙ КОД

```
from graph import *  
windowSize(640, 480)  
canvasSize(640, 480)  
x = 20  
while x <= 620:  
    circle(x, 240, 20)  
    x = x + 40
```



Измените программу так, чтобы:

- рисовались круги радиусом 30;
- рисовался вертикальный ряд кругов.

# ЧИТАЕМ ПРОГРАММНЫЙ КОД

```
x = int(input())
y = int(input())
while x != y:
    if x > y:
        x = x - y
    else:
        y = y - x
nod = x
print('НОД = ', nod)
```

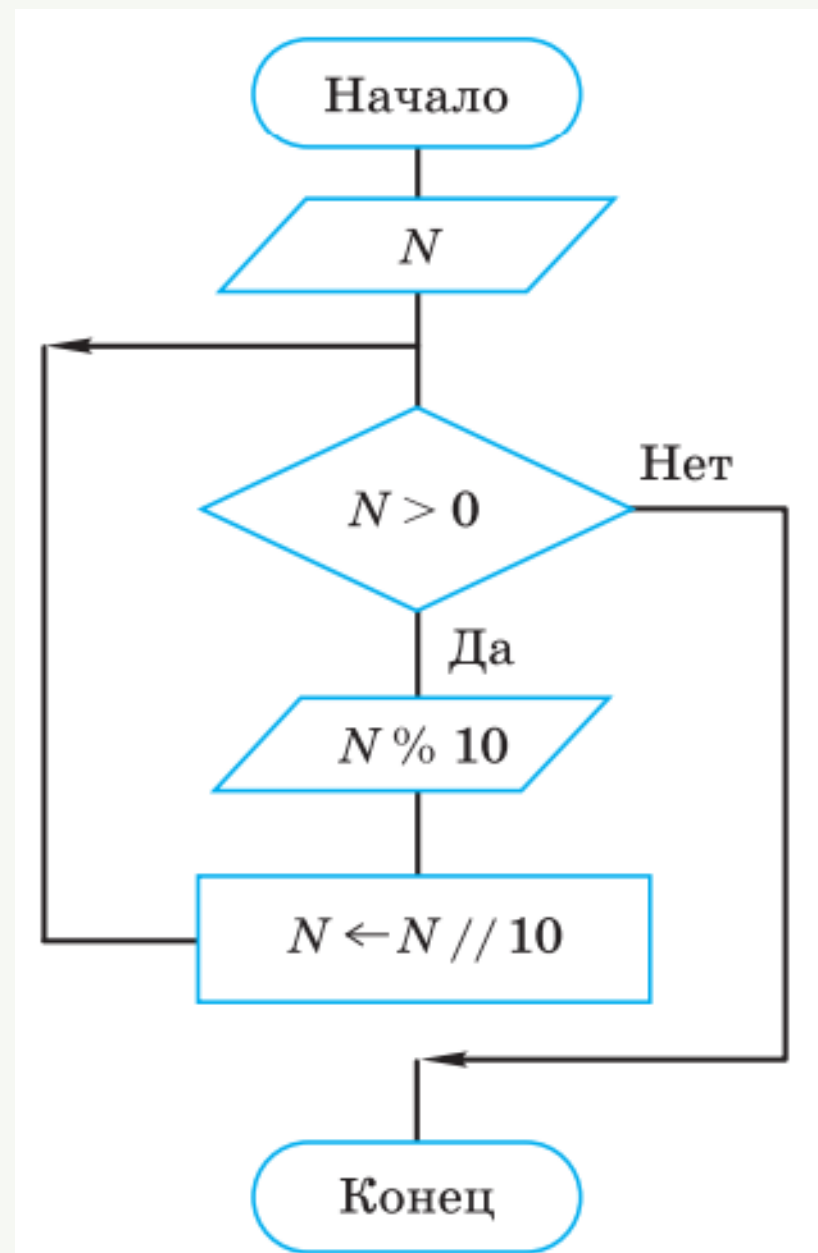
X	Y	X!=Y	X>Y
450	180	Да	Да
270	180	Да	Да
90	180	Да	Нет
90	90	Нет	

**Алгоритм Евклида**

# ЧИТАЕМ ПРОГРАММНЫЙ КОД

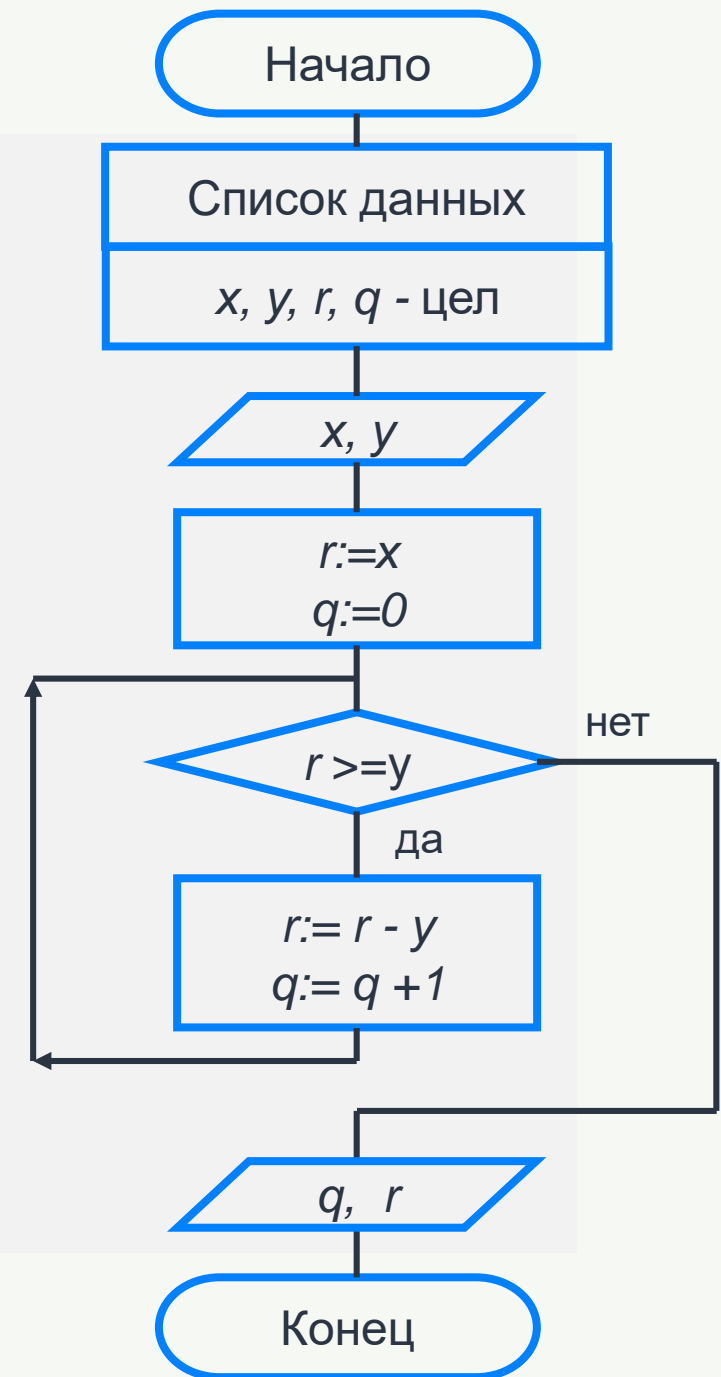
```
n = int(input())  
while n > 0:  
    print(n % 10)  
    n = n // 10
```

Что получится, если вместо 10  
использовать 2; 5; 8?





```
print ('Частное и остаток')
x = int(input('Введите делимое x>>'))
y = int(input('Введите делитель y>>'))
r = x
q = 0
while r >= y:
    r = r - y
    q += 1
print ('Частное q=', q)
print ('Остаток r=', r)
```



# ЧИТАЕМ ПРОГРАММНЫЙ КОД

12, 13, 18, 23, 120, 83, 113, 0

```
k = 0
a = int(input())
while a != 0:
    if a % 10 == 3:
        k += 1
    a = int(input())
print('k=', k)
```

**k = 4**

```
print('Обработка последовательности')
k = 0
print('Первый член последовательности>>')
a = int(input())
while a != 0:
    if a % 10 == 3:
        k += 1
    print('Очередной член последовательности или 0>>')
    a = int(input())
print('k=', k)
```

# ПРОГРАММИРОВАНИЕ ЦИКЛОВ С ЗАДАНЫМ УСЛОВИЕМ ОКОНЧАНИЯ РАБОТЫ

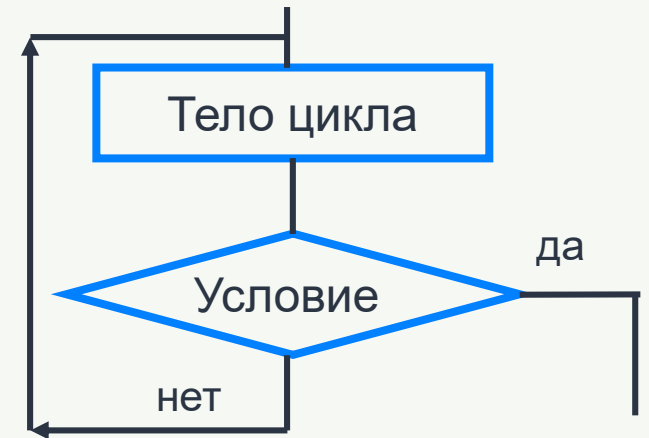
В языке Python нет цикла с заданным условием окончания работы, но его можно организовать с помощью цикла **while**:

```
while True:  
    <операторы>  
    if <условие>: break
```

Цикл

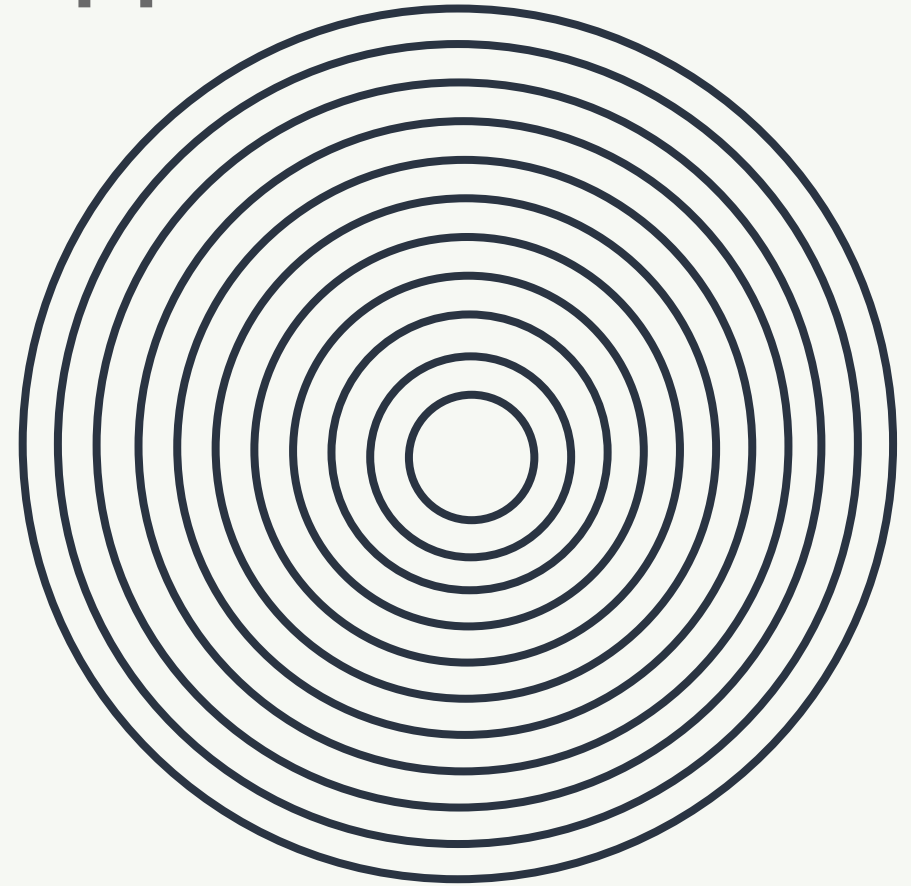
```
while True:  
    <операторы>
```

будет выполняться бесконечно, потому что условие `True` всегда истинно. Выйти из такого цикла можно только с помощью специального оператора **break** (англ. - прервать).



# ЧИТАЕМ ПРОГРАММНЫЙ КОД

```
from graph import *  
windowSize(640, 480)  
canvasSize(640, 480)  
r = 200  
while True:  
    circle(320, 240, r);  
    r = r - 5  
    if r < 10: break
```



Попробуйте получить такое же изображение, используя в программе цикл с заданным условием продолжения работы.

# ЧИТАЕМ ПРОГРАММНЫЙ КОД

```
n = int(input('Введите число n>>'))  
k = 0  
while True:  
    n = n // 10  
    k += 1  
    if n == 0: break  
print ('k = ', k)
```

125 → 3

```
print ('Подсчёт цифр в числе')
n = int(input('Введите число n>>'))
k = 0
while True:
    n = n // 10
    k += 1
    if n == 0: break
print ('k = ', k)
```



Предложите решение этой задачи с использованием цикла с заданным условием продолжения работы.

Доработайте программу так, чтобы в ней находилась ещё и сумма цифр исходного числа.

# ЧИТАЕМ ПРОГРАММНЫЙ КОД

```
i = 1
x = 10
while True:
    i += 1
    x = x + 0.1 * x
    if x >= 25: break
print ('i=', i)
```





```
print ('График тренировок')
i = 1
x = 10
while True:
    i += 1
    x = x + 0.1 * x
    if x >= 25: break
print ('Начиная с ', i, '-го дня спортсмен будет
пробежать 25 км')
```

# ПРОГРАММИРОВАНИЕ ЦИКЛОВ С ФИКСИРОВАННЫМ ЧИСЛОМ ПОВТОРЕНИЙ

Общий вид оператора:

```
for <параметр> in range (k, n, m):  
    <оператор>
```

<параметр> - переменная целого типа

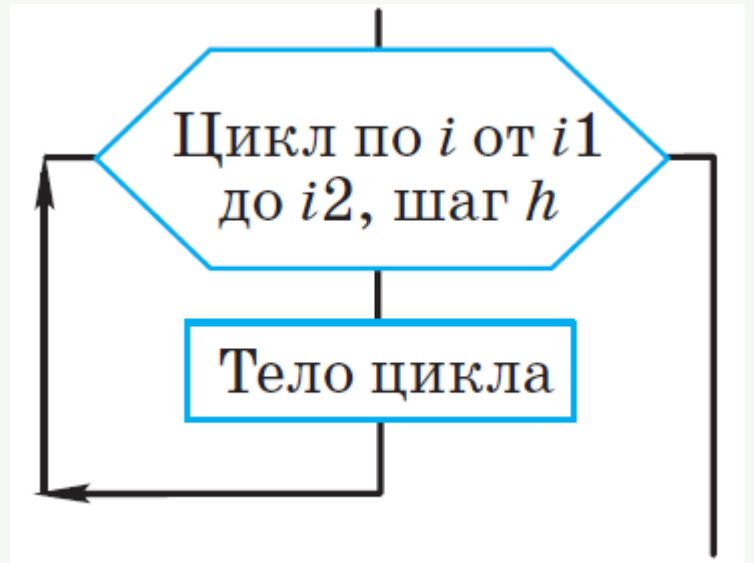
range() – диапазон значений:

**k** – начальное значение переменной, по умолчанию равно 0

**n** – конечное значение переменной, не включая последнее

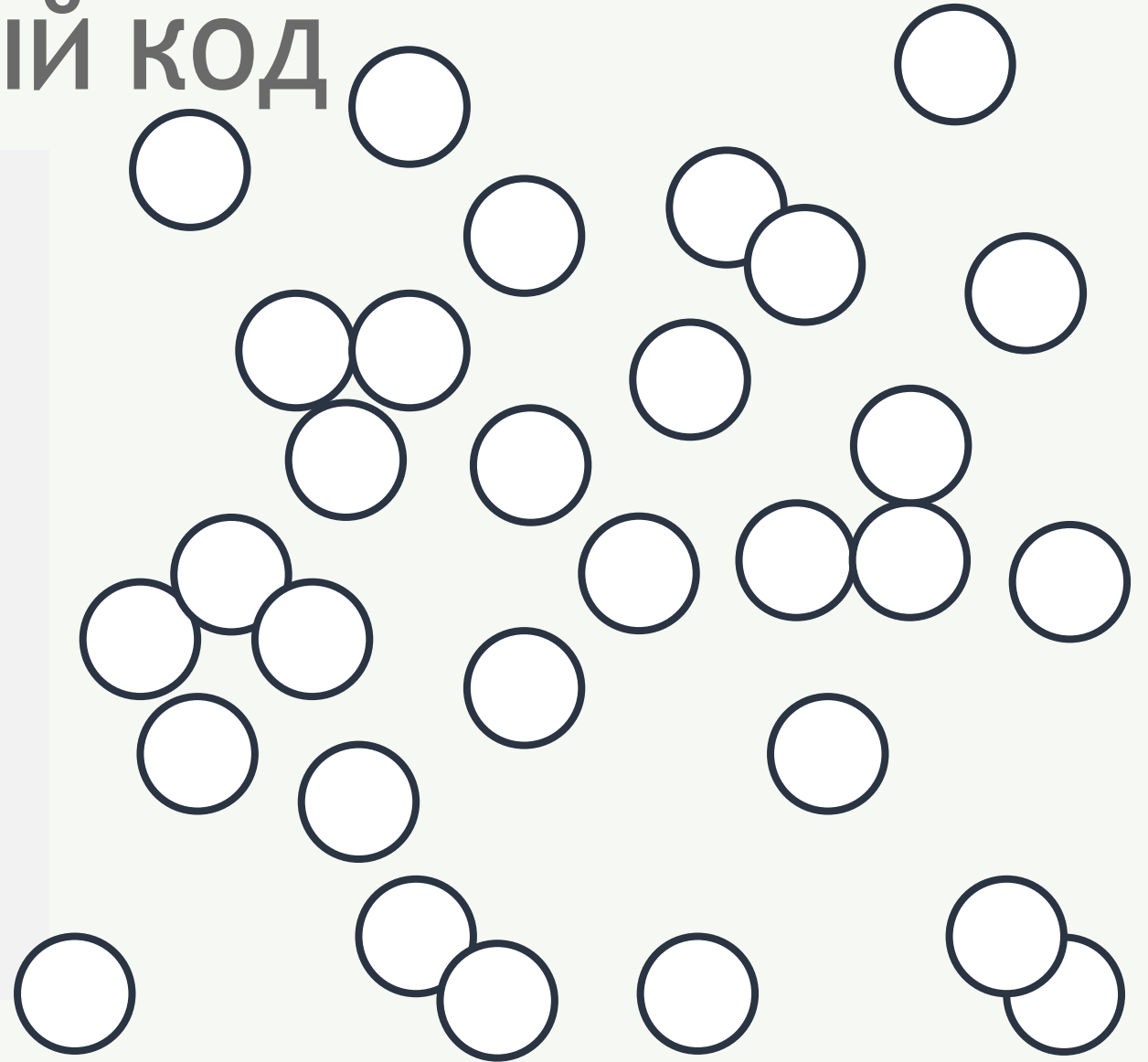
**m** – шаг изменения переменной, по умолчанию равен 1

После каждого выполнения тела цикла происходит увеличение на единицу параметра цикла; условие выхода из цикла - достижение параметром конечного значения.



# ЧИТАЕМ ПРОГРАММНЫЙ КОД

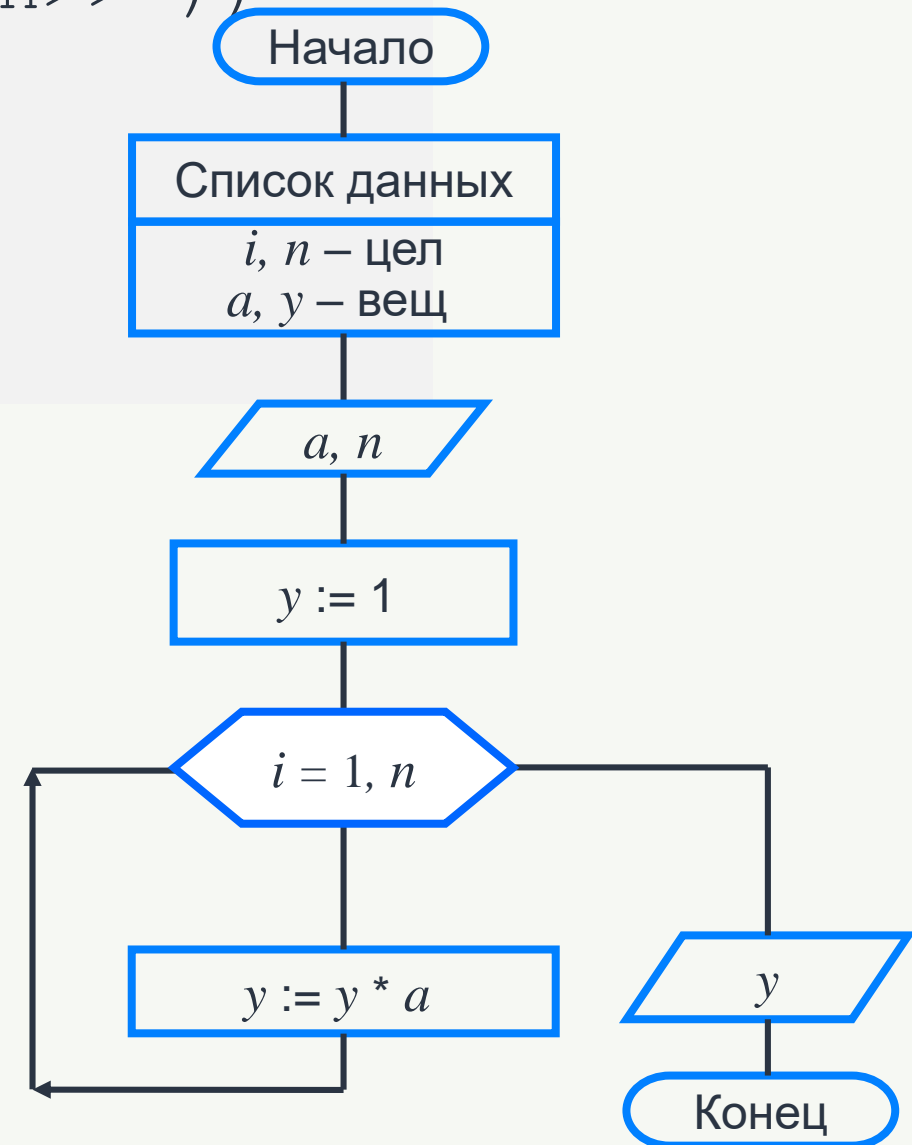
```
from math import *
from graph import *
windowSize(640, 480)
canvasSize(640, 480)
for i in range(100):
    x = randint(1, 640)
    y = randint(1, 480)
    circle(x, y, 10)
```



Внесите изменения в программу так, чтобы радиусы изображаемых кругов также генерировались случайным образом.

```
# Возведение в степень
a = float(input('Введите основание a>>'))
n = int(input('Введите показатель n>>'))
y = 1
for i in range(n):
    y = y * a
print('y=', y)
```

$a^n$



# ВЛОЖЕННЫЕ ЦИКЛЫ

Цикл называется **вложенным**, если он содержится внутри (в теле) другого цикла.

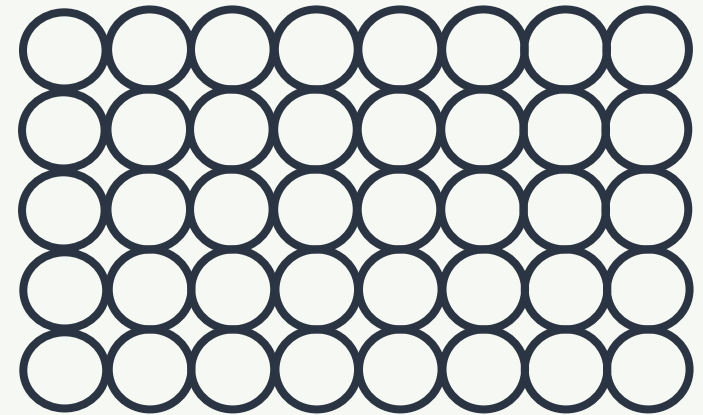
Цикл, содержащий в себе другой цикл, называют **внешним**, а цикл, содержащийся в теле другого цикла, — **внутренним**.

Внутренний и внешний циклы могут быть любыми из трёх видов: цикл с переменной, цикл с предусловием или цикл с постусловием.



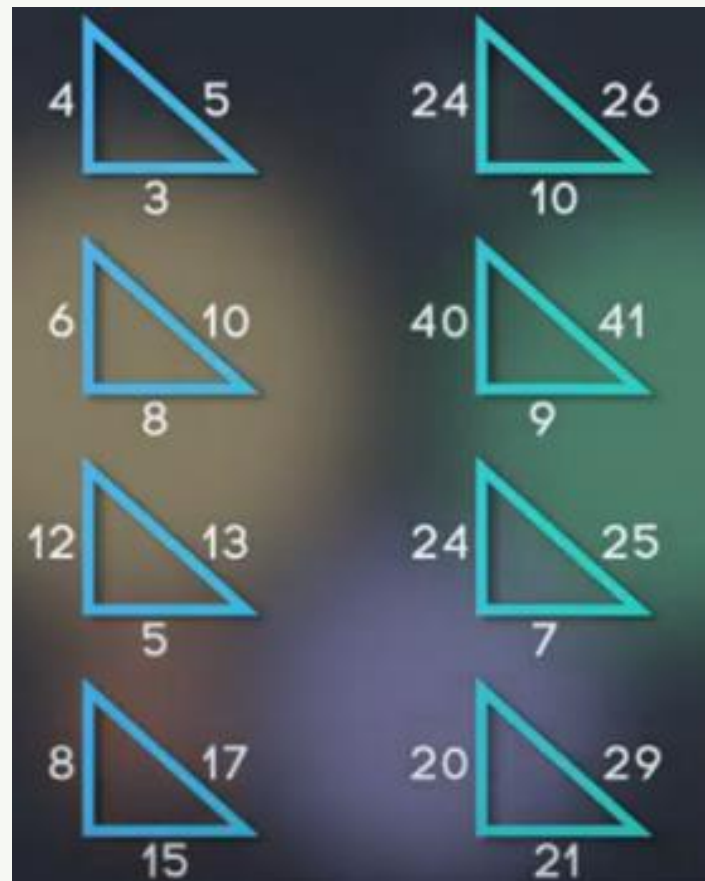
# ЧИТАЕМ ПРОГРАММНЫЙ КОД

```
from graph import *  
windowSize(640, 480)  
canvasSize(640, 480)  
y = 10  
for i in range(5):  
    x = 10  
    for j in range(8):  
        circle(x, y, 10)  
        x += 20  
    y += 20
```



# ЧИТАЕМ ПРОГРАММНЫЙ КОД

```
for a in range (1, 51):  
    for b in range (a, 51):  
        x = a * a + b * b  
        c = 1  
        while c * c <= x:  
            if c * c == x:  
                print(a, b, c)  
            c = c + 1
```

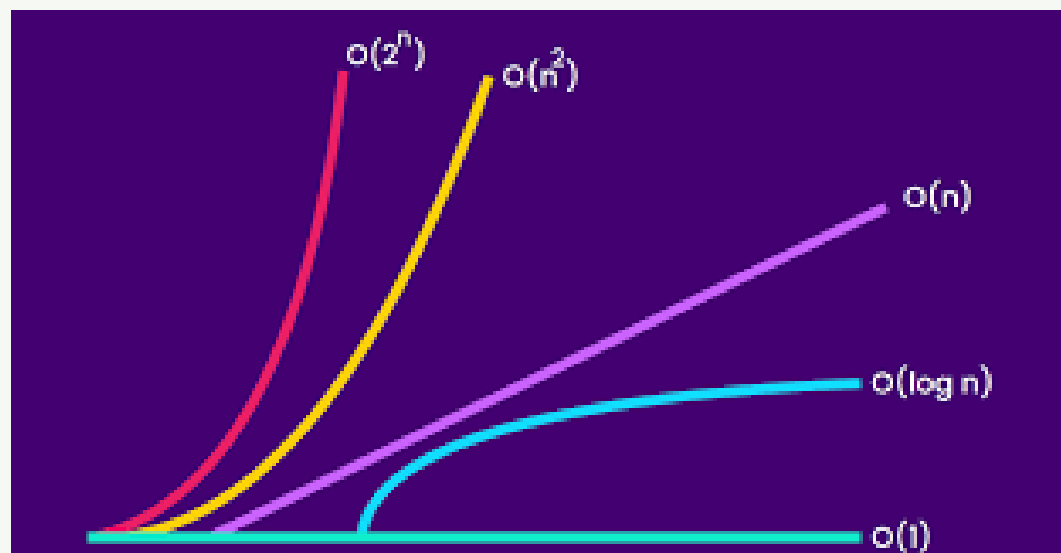


# СЛОЖНОСТЬ АЛГОРИТМА

Алгоритм состоит из команд. **Команда** — это отдельная инструкция в описании алгоритма. **Шаг** алгоритма — это отдельное действие, которое исполнитель выполняет по команде.

**Сложность алгоритма** — это количество элементарных шагов (действий), совершённых при исполнении этого алгоритма.

Алгоритм, выполняющий только операции чтения данных и занесения их в оперативную память, имеет линейную сложность  $O(n)$  (читается «о большое от эн»). Существуют алгоритмы, имеющие квадратичную и кубическую сложности.





**Оператор цикла** – команда, реализующая алгоритмическую конструкцию «повторение» на языке программирования.

В языке программирования Python существует несколько операторов цикла:

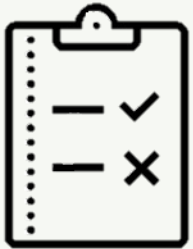
- ◆ **while** (цикл с условием),
- ◆ **for** (цикл с параметром).

Цикл с условием выполняется до тех пор, пока некоторое условие не станет ложным.

Если условие в заголовке цикла всегда остаётся истинным, цикл работает бесконечно. Для досрочного выхода из цикла используют оператор `break`.

Цикл с параметром применяют тогда, когда количество повторений цикла известно заранее или может быть вычислено до начала цикла.

**Сложность алгоритма** – это количество элементарных шагов (действий), совершённых при исполнении этого алгоритма



# ВОПРОСЫ И ЗАДАНИЯ

Дана последовательность операторов:

```
a = 1
```

```
b = 2
```

```
while a + b < 8:
```

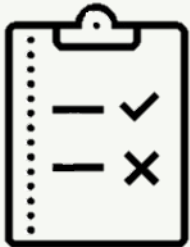
```
    a = a + 1
```

```
    b = b + 2
```

```
s = a + b
```

Сколько раз будет повторен цикл и какими будут значения переменных **a**, **b**, **s** после исполнения этой последовательности операторов?





## ВОПРОСЫ И ЗАДАНИЯ

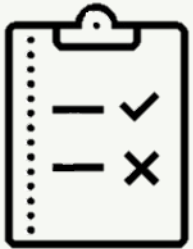
Требовалось написать программу вычисления факториала числа  $n$  (факториал числа  $n$  есть произведение всех целых чисел от 1 до  $n$ ).

Программист торопился и написал программу неправильно. Ниже приведён фрагмент его программы с ошибками:

```
k = 1
f = 0
while k < n:
    f = f * k
    k += 1
```

Найдите ошибки. Допишите необходимые операторы.





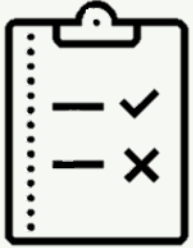
# ВОПРОСЫ И ЗАДАНИЯ

Дана последовательность операторов:

```
a = 1
b = 1
while True:
    a = a + 1
    b = b * 2
    if b > 8: break
s = a + b
```

Сколько раз будет повторено тело цикла и какими будут значения переменных **a**, **b**, **s** после исполнения этой последовательности операторов?

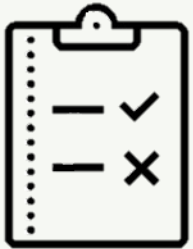




# ВОПРОСЫ И ЗАДАНИЯ

Напишите программу, в которой осуществляется ввод целых чисел (ввод осуществляется до тех пор, пока не будет введён ноль) и определение максимального (наибольшего) из введённых чисел.





# ВОПРОСЫ И ЗАДАНИЯ

Сколько раз будет выполнено тело цикла?

а) `for i in range(15): s = s + 1`

б) `for i in range(10, 15): s = s + 1`

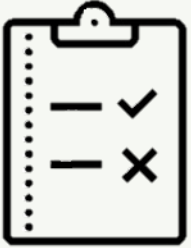
в) `for i in range(-1, 1): s = s + 1`

г) `for i in range(1, 1): s = s + 1`

д) `k = 5`

`for i in range(k-1, k+1): s = s + 1`

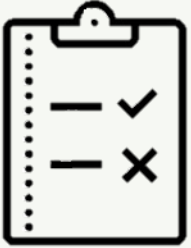




# ВОПРОСЫ И ЗАДАНИЯ

Напишите программу, которая 10 раз выводит на экран ваши имя и фамилию.





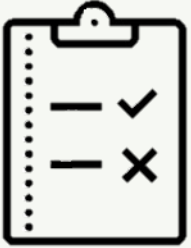
# ВОПРОСЫ И ЗАДАНИЯ

Напишите программу, выводящую на экран изображение шахматной доски, где чёрные клетки изображаются звёздочками, а белые - пробелами. Рекомендуемый вид экрана после выполнения программы представлен ниже:

```
*      *      *      *
 *     *     *     *     *
*     *     *     *     *
 *     *     *     *     *
*     *     *     *     *
 *     *     *     *     *
*     *     *     *     *
```





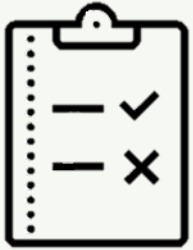


# ВОПРОСЫ И ЗАДАНИЯ

Напишите программу, которая вычисляет сумму:

- а) первых  $n$  натуральных чисел;
- б) квадратов первых  $n$  натуральных чисел;
- в) всех чётных чисел в диапазоне от 1 до  $n$ ;
- г) всех двузначных чисел.





# ВОПРОСЫ И ЗАДАНИЯ

Напишите программу, которая выводит на экран таблицу умножения на  $n$  ( $n$  - целое число в диапазоне от 2 до 10, вводимое с клавиатуры).

Пример входных данных	Пример выходных данных
Введите $n > 5$	$5 * 2 = 10$ $5 * 3 = 15$ $5 * 4 = 20$ $5 * 5 = 25$ $5 * 6 = 30$ $5 * 7 = 35$ $5 * 8 = 40$ $5 * 9 = 45$ $5 * 10 = 50$





# ВОПРОСЫ И ЗАДАНИЯ

Напишите программу, которая выводит на экран таблицу степеней двойки (от нулевой до десятой).

Рекомендуемый вид экрана после выполнения программы представлен ниже:

Таблица степеней двойки:

0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

В языке Python имеются операторы цикла:

*for* (цикл с параметром)

Число повторений  
цикла известно

*while* (цикл-ПОКА)

Число повторений  
цикла неизвестно