

КУРС «ИНФОРМАТИКА»

Часть 4. Алгоритм разбора целого числа
на цифры

2023 – 2024 УЧЕБНЫЙ ГОД

Операции % и //

$$7 // 3 = 2$$

$$8 // 2 = 4$$

$$1 // 3 = 0$$

$$7 \% 3 = 1$$

$$10 \% 4 = 2$$

$$3 \% 5 = 3$$

% 10 – остаток от деления на 10 любого числа,
позволяет определить младшую (правую) цифру числа:

$$137 \% 10 = 7$$

$$2134 \% 10 = 4$$

// 10 – целая часть от деления

Операции, определяющей старшую цифру числа **нет!**

Сумма цифр числа

Программный код

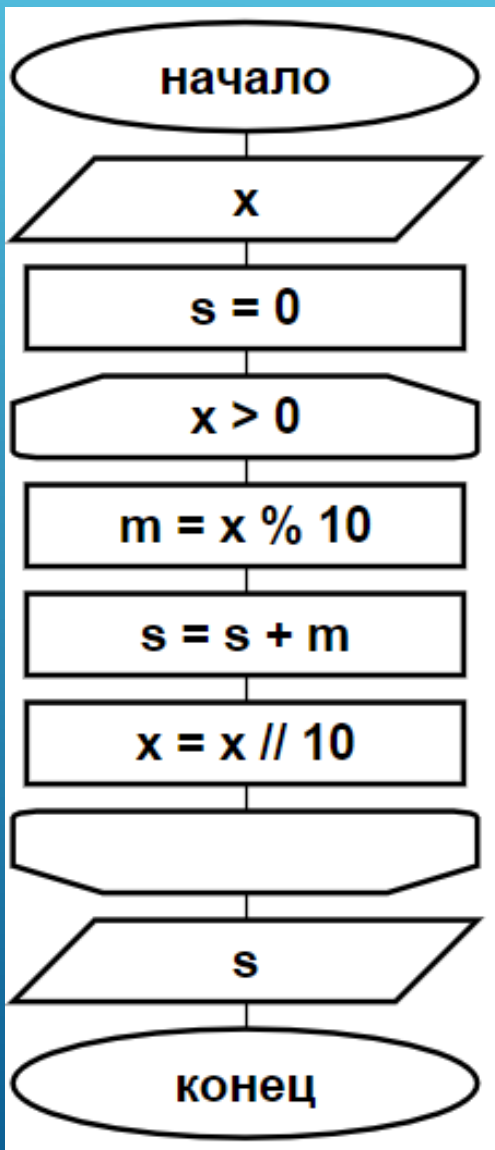
```
x = int(input("Введите число"))
s = 0
while x > 0:
    m = x % 10
    s = s + m
    x = x // 10
print(s)
```

Пример

Найти сумму
цифр числа 123

Трассировка алгоритма

x	m	s
123	3	3
12	2	5
1	1	6
0		



Сумма цифр числа

Программный код

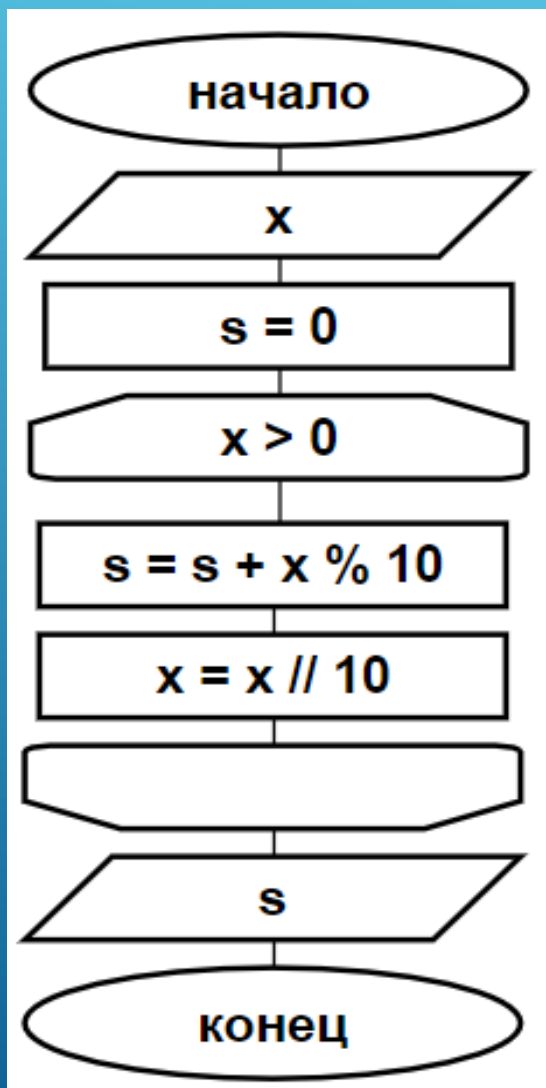
```
x = int(input("Введите число"))  
s = 0  
while x > 0:  
    s = s + x % 10  
    x = x // 10  
print(s)
```

Пример

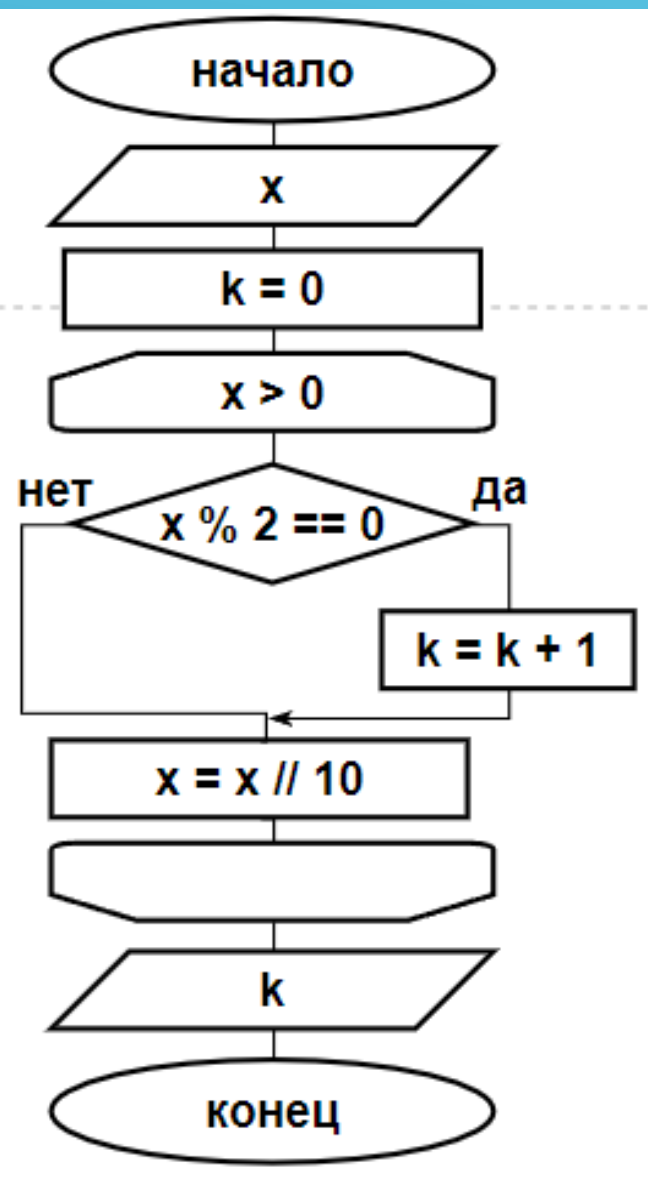
Найти сумму
цифр числа 123

Трассировка
алгоритма

x	s
123	3
12	5
1	6
0	



Количество четных цифр числа



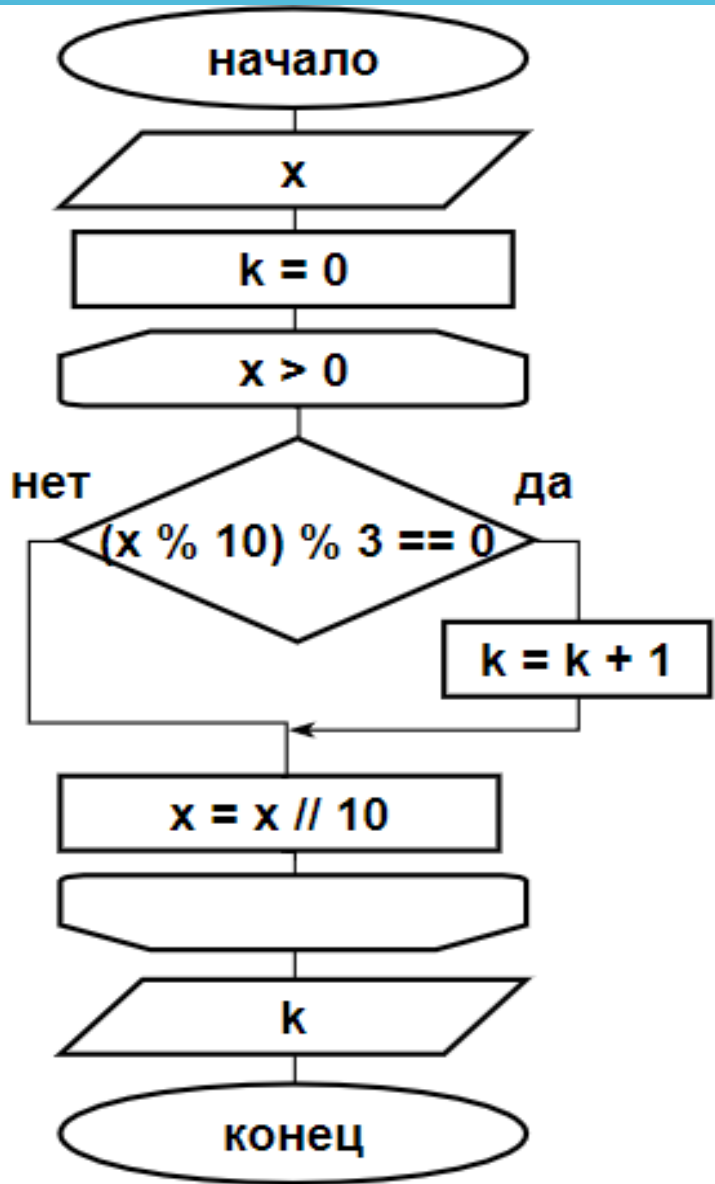
```
x = int(input("Введите число "))
k = 0
while x > 0:
    if x % 2 == 0:
        k = k + 1
    x = x // 10
print(k)
```

можно проверять
НА ЧЕТНОСТЬ
не только последнюю
цифру, а все число

$$(x \% 10) \% 2 \equiv x \% 2$$

Количество цифр числа, кратных 3

6



```
x = int(input("Введите число "))
k = 0
while x > 0:
    if (x % 10) % 3 == 0:
        k = k + 1
    x = x // 10
print(k)
```

это правильно

нельзя проверять на
кратность не последнюю
цифру, а все число

$$(x \% 10) \% 3 \neq x \% 3$$

$$(53 \% 10) \% 3 \neq 53 \% 3$$

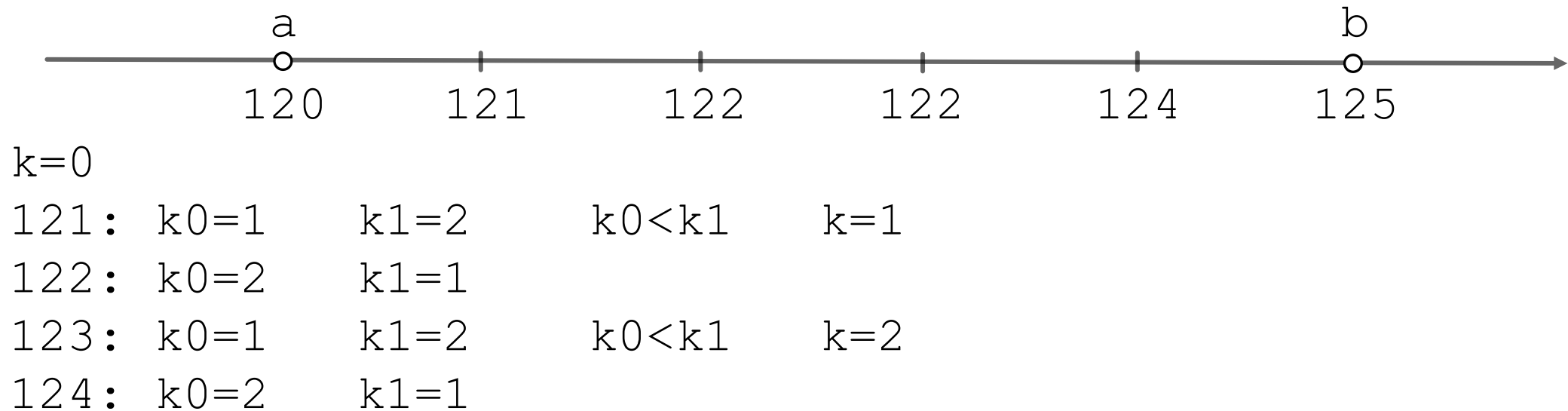
КУРС «ИНФОРМАТИКА»

Часть 5. Подготовка
к контрольной работе № 2

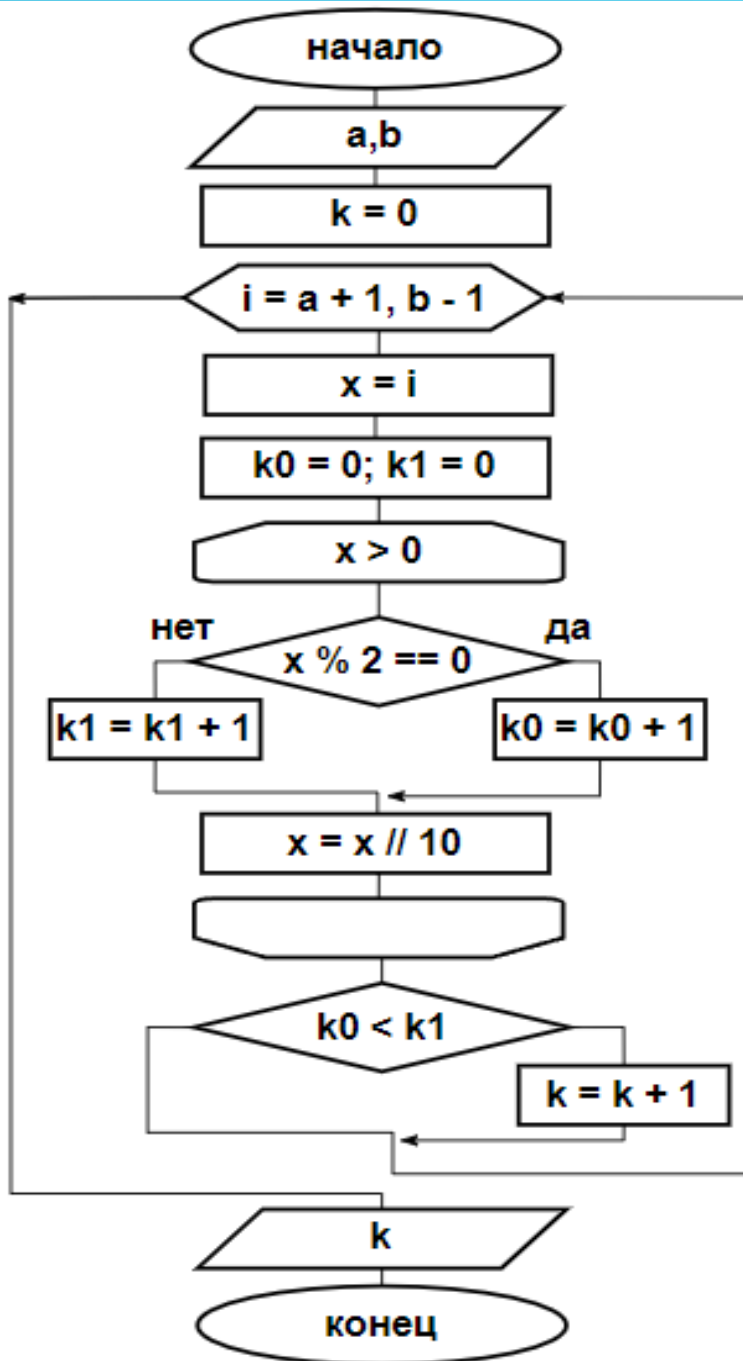
2023 – 2024 УЧЕБНЫЙ ГОД

Пример задачи 1

1.1. Составить алгоритм и написать (с использованием функции) программу подсчета количества натуральных чисел, принадлежащих интервалу (a, b) , у которых количество четных цифр меньше количества нечетных цифр.

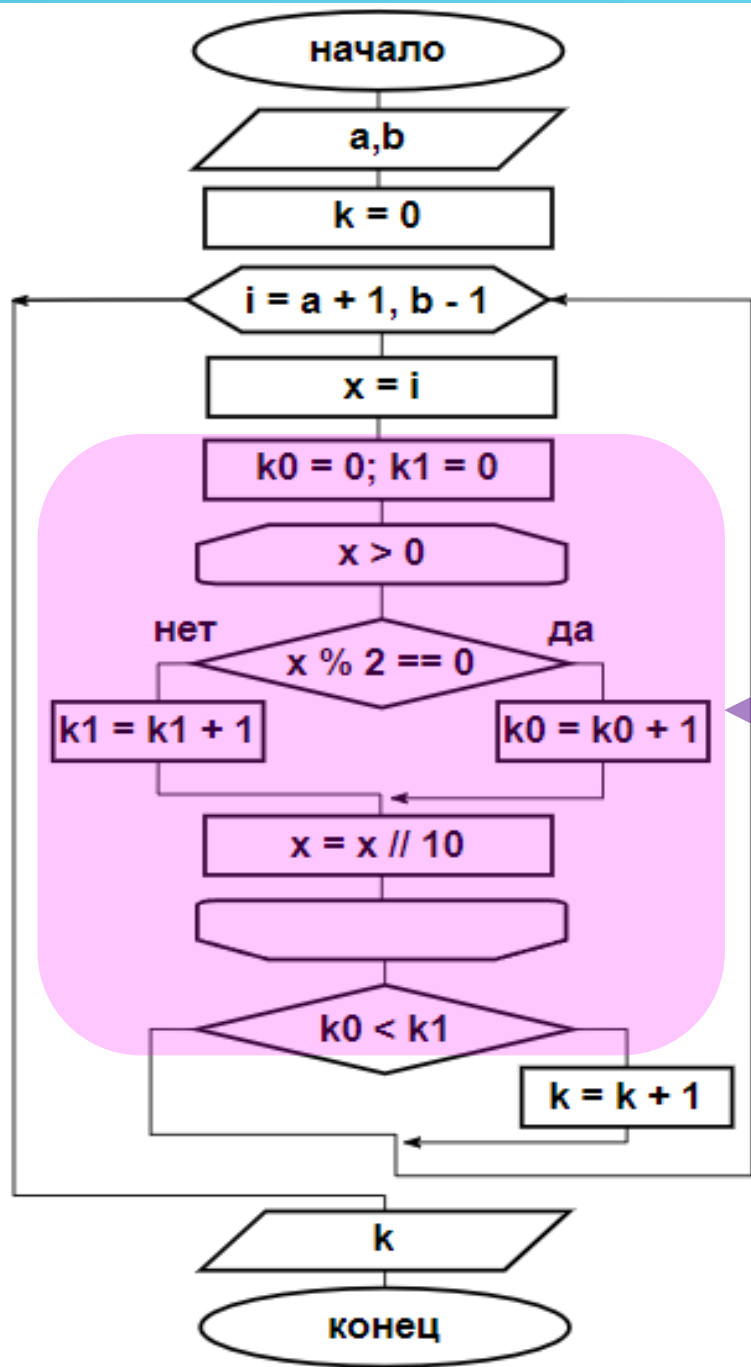


Задача 1.1



1. Ввод a и b .
2. $k=0$ – общее количество подходящих чисел
3. Цикл по интервалу (a, b)
 - 3.1. Дублируем переменную цикла
 - 3.2. $k0$ – счетчик четных, $k1$ – счетчик нечетных
 - 3.3. Цикл по числу
 - 3.3.1. Проверка четности
 - 3.3.1.1. Увеличение счетчиков
 - 3.3.2. Деление на 10
 - 3.4. Проверка условия задачи: $k0 < k1$
 - 3.4.1. Увеличение счетчика k
4. Вывод k

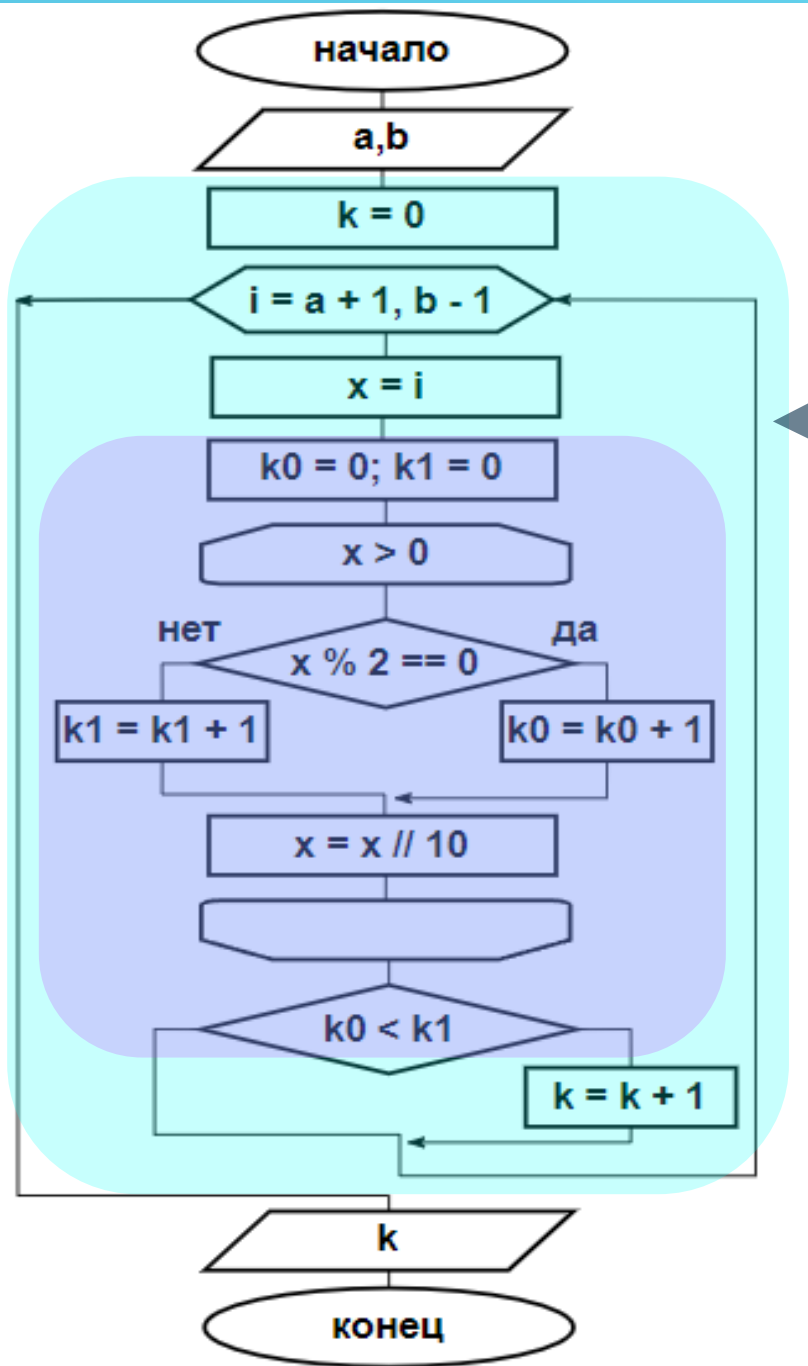
Задача 1.1



```

def test(x):
    k0 = 0
    k1 = 0
    while x > 0:
        if x % 2 == 0:
            k0 = k0 + 1
        else:
            k1 = k1 + 1
        x = x // 10
    return (k0 < k1)
  
```

Задача 1.1

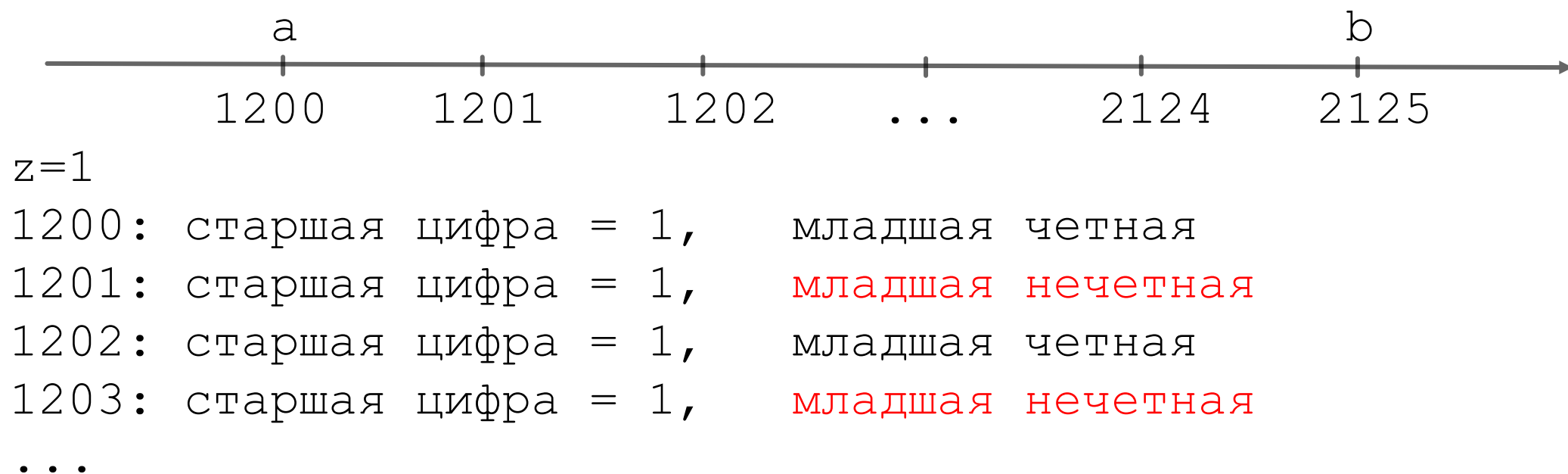


```
def alg(a,b):
    k = 0
    for i in range(a+1, b):
        if test(i):
            k += 1
    return k
```

```
a,b = map(int,input("Введите границы")
              .split())
print("Количество= ", alg(a,b))
```

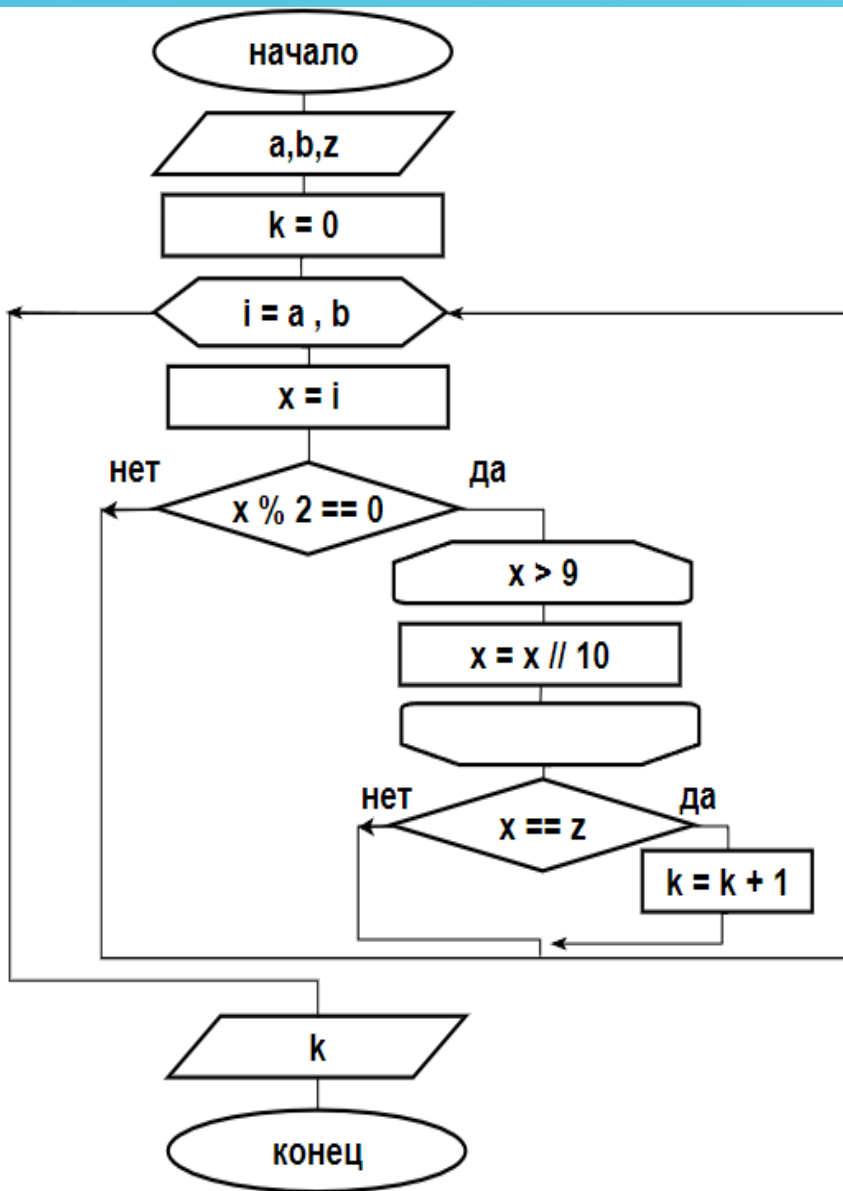
Пример задачи 1

1.2. Составить алгоритм и написать (с использованием функции) программу подсчета количества натуральных чисел, принадлежащих отрезку $[a, b]$, у которых старшая цифра равна заданной цифре, а младшая четная.



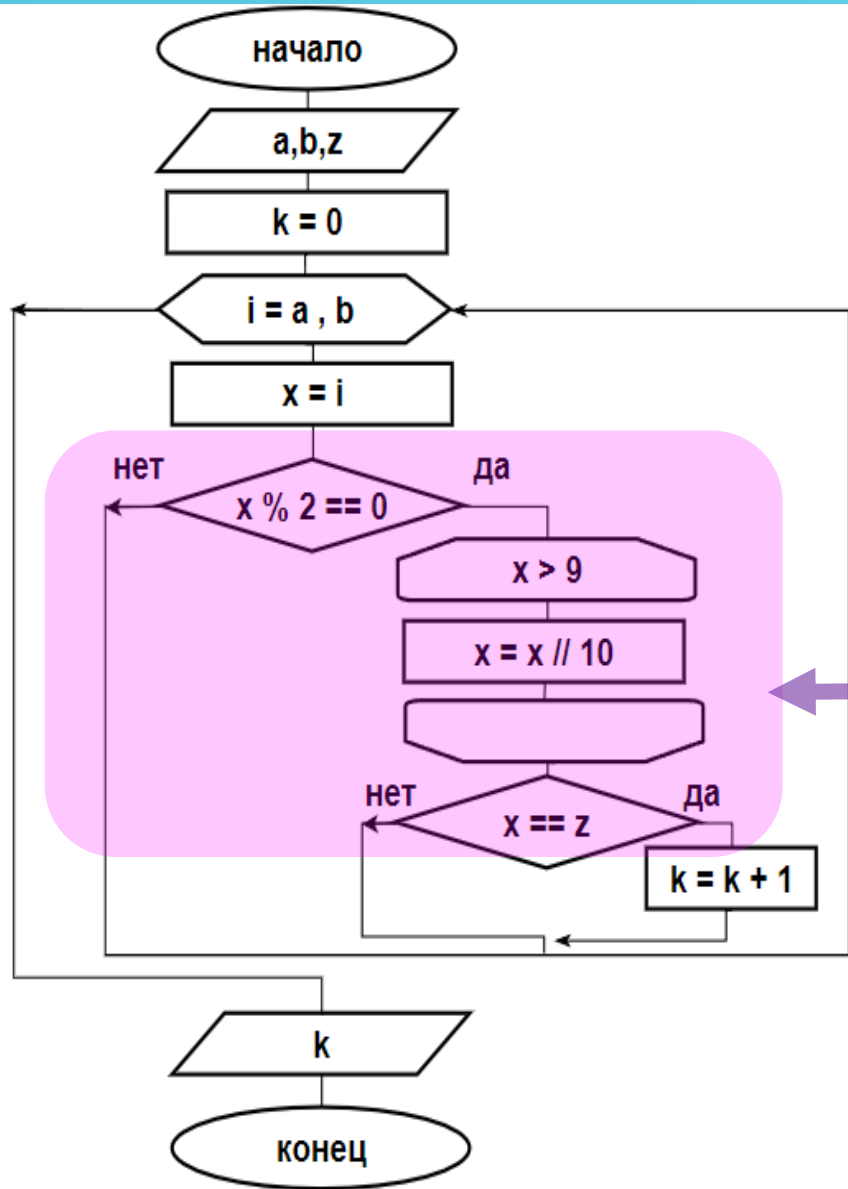
Задача 1.2

13



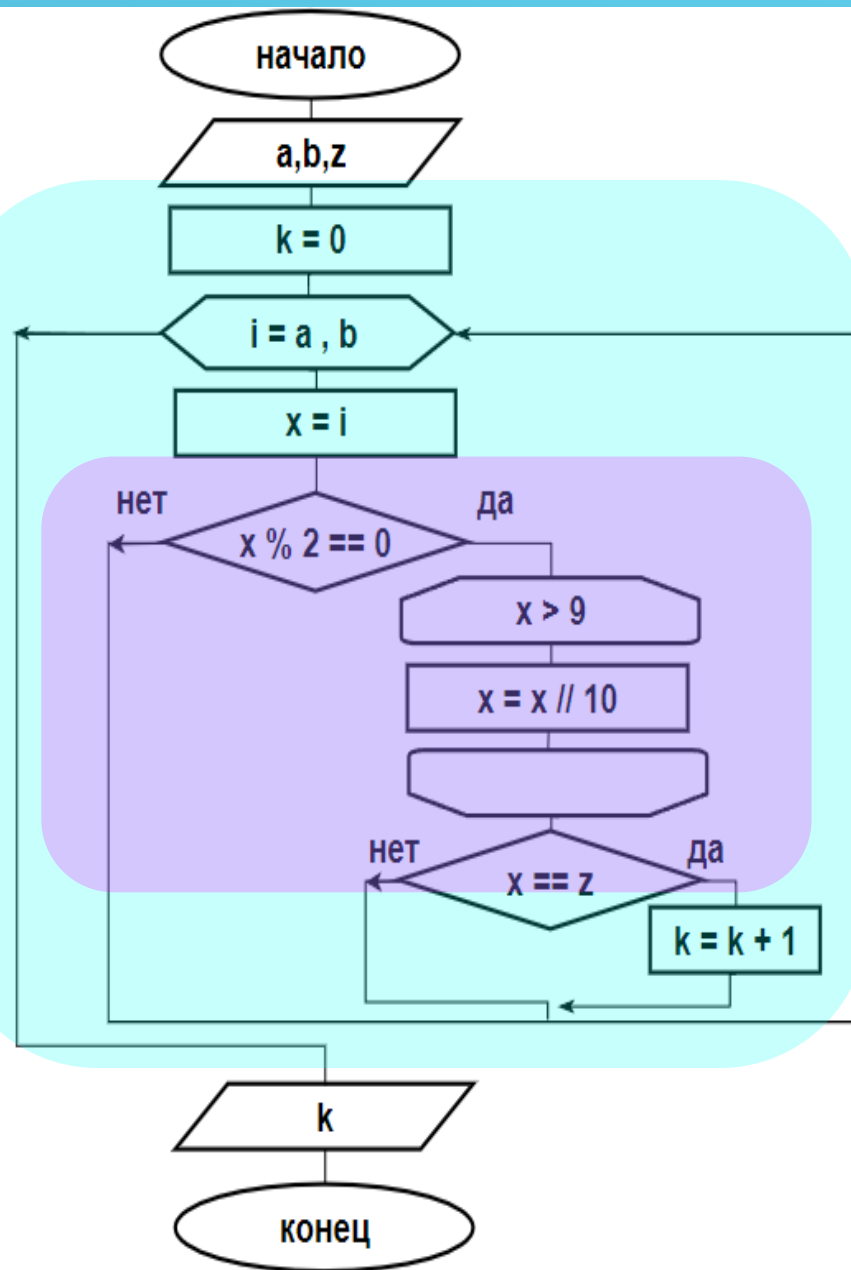
1. Ввод a, b (границы) и z (заданное число).
2. $k=0$ – общее количество подходящих чисел
3. Цикл по отрезку $[a, b]$
 - 3.1. Проверка условия четности числа i
 - 3.1.1. Дублируем переменную цикла
 - 3.1.2. Цикл по числу (поиск старшей цифры)
 - 3.1.2.1. Деление на 10
 - 3.1.3. Проверка условия задачи: $x = z$
 - 3.1.3.1 Увеличение счетчика k
4. Вывод k

Задача 1.2



```
def test(x, z):
    if (x%2==0):
        while x > 9:
            x = x // 10
        return x==z
    else:
        return False
```

Задача 1.2



```
def alg(a,b,z):
    k = 0
    for i in range(a, b+1):
        if test(i,z):
            k += 1
    return k
```

```
a,b,z =
map(int,input("Введите границы и
цифру ").split())
print("Количество= ", alg(a,b,z))
```

Пример задачи 2

2.1 Написать (с использованием функции) программу подсчета количества четных чисел Фибоначчи, принадлежащих отрезку $[a, b]$, у которых сумма цифр – четная.

```
def test(x):  
    if(x % 2 == 0):  
        s = 0  
        while x > 0:  
            s = s + x % 10  
            x = x // 10  
        return (s % 2 == 0)  
    else:  
        return False
```


Пример задачи 2

Числа Фибоначчи

1	1	2	3	5	8	13	...
f1	f2	f					

```
def fib(a,b):  
    f1 = 1  
    f2 = 1  
    k = 0  
    while (f1 <= b):  
        if (f1 >= a) and test(f1):  
            k = k + 1  
        f = f1 + f2  
        f1 = f2  
        f2 = f  
    return k
```

Пример задачи 2

```
a,b = map(int,input("Введите границы").split())  
print("Количество= ", fib(a,b))
```

Пример задачи 2

2.2 Написать (с использованием функции) программу подсчета количества чисел Фибоначчи, принадлежащих отрезку $[a, b]$, в которых все цифры одинаковые.

1 1 2 3 5 8 13 21 34 55 89...

```
def test(x):
    m = x % 10
    while x > 0 and x % 10 == m:
        x = x // 10
    return x == 0
```

```
m = 2

22222      x == 0

22122      x != 0
```

Пример задачи 2

20

```
def fib(a,b):  
    f1 = 1  
    f2 = 1  
    k = 0  
    while (f1 <= b):  
        if (f1 >= a) and test(f1):  
            k = k + 1  
        f = f1 + f2  
        f1 = f2  
        f2 = f  
    return k  
  
a,b = map(int,input("Введите границы ").split())  
print("Количество= ", fib(a,b))
```


Дополнительные задачи

для подготовки к КР № 2

- | | |
|--|---|
| 1 Составить алгоритм и написать (с использованием функции) программу подсчета количества натуральных чисел, принадлежащих интервалу (a, b) , у которых количество цифр, не кратных 3, равно заданному числу. | 2 Написать (с использованием функции) программу подсчета количества четных чисел Фибоначчи, принадлежащих отрезку $[a, b]$, у которых количество четных цифр кратно количеству нечетных. |
| 1 Составить алгоритм и написать (с использованием функции) программу подсчета количества натуральных чисел, принадлежащих интервалу $[a, b)$, в которых сумма цифр нечетная и некрatна младшей цифре. | 2 Написать (с использованием функции) программу подсчета количества чисел Фибоначчи, принадлежащих отрезку $(a, b]$, у которых сумма младшей и старшей цифр кратна заданному числу |

Источники

Обзорный материал (данная презентация)

 КУРС «ИНФОРМАТИКА». Практика 2. Целочисленная арифметика.pptx

КУРС «ИНФОРМАТИКА»

Часть 6. Памятка по Python № 2

2023 – 2024 УЧЕБНЫЙ ГОД

Создание функции

Описание	Пример
Функция в python – объект, принимающий аргументы и возвращающий значение. Обычно функция определяется с помощью инструкции def .	<pre>def add(x, y): return x + y #данная функция возвращает сумму чисел</pre>
Для вызова функции необходимо указать ее имя и ее аргументы	<pre>print(add(2, 3)) #выводит на консоль 5</pre>
Функция может и не заканчиваться инструкцией <code>return</code> , при этом функция вернет значение <u>None</u>	<pre># задание функции def foo(): print("Hello, world") #вызов функции foo()</pre>

Список функций

Название функции	Описание	Пример использования
<code>map(функция, последовательность1,.....)</code>	Применяет указанную функцию к каждому элементу указанной последовательности/последовательностей	<pre># конвертирует две строки в числа x,y = map(int, ["5","2"]) print(x,y)</pre>
<code>split(separator*, maxsplit*)</code>	Разделяет основную строку по разделителю и возвращает список строк. Параметр <code>separator</code> –необязательный, он позволяет задать разделитель вручную. По умолчанию разделитель – пробел Параметр <code>maxsplit</code> определяет максимальное количество разделений (по умолчанию выполняет все деления)	<pre>my_st = "Пример строки Python" print(my_st.split()) # my_st разбилась на 3 #отдельные строки "Пример", #"строки", "Python"</pre>