

Алгоритмы и структуры данных на Python

Урок 1



Введение в алгоритмизацию и реализация простых алгоритмов на Python.

Введение в алгоритмизацию. Решение
практических задач.

Регламент урока

1. Время урока - 1.5 часа
2. Домашние задания
3. Видеозапись будет
4. Вопросы – в специальное время.



Вопросы

1. Что такое алгоритмизация?
2. Способы представления алгоритмов.
3. Разбор простых алгоритмов.



Цели урока

1. Изучить основные понятия алгоритмизации
1. Изучить способы представления алгоритмов
1. Практика разработки простых алгоритмов



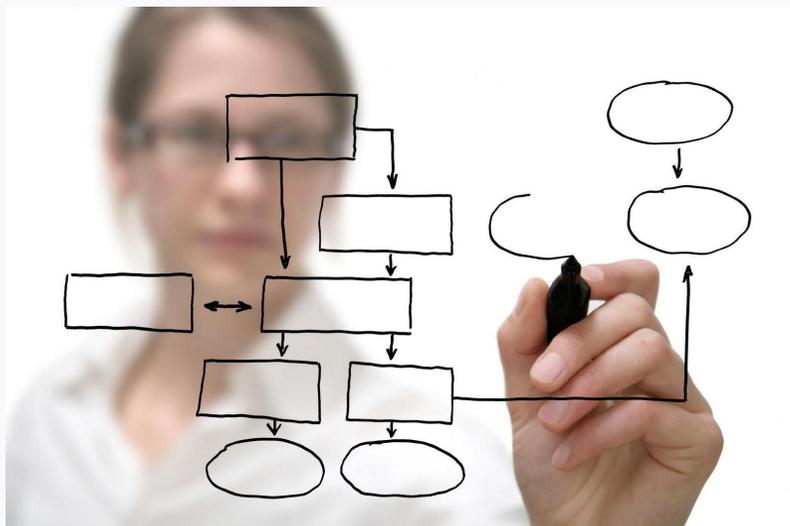
План урока

	Часть 1	Часть 2
Урок 1	Основы алгоритмизации. Блок-схемы. Основные конструкции блок-схем.	Практика



Понятие алгоритма

Алгоритм – это последовательность чётко определенных действий, выполнение которых ведёт к решению задачи.



Каким должен быть алгоритм



Блок-схема

Таблица 1.2. Базовые элементы блок-схемы		
Название	Обозначение	Функция
Терминатор		Отображает вход из внешней среды или выход из нее (наиболее частое применение – обозначение начала и конца алгоритма)
Процесс		Выполнение одной или нескольких операций, обработка данных любого вида, изменение данных. Внутри фигуры записывают операции над данными.
Решение		Отображает решение, влияющее на дальнейшее выполнение алгоритма. Используется для обозначения альтернативных ветвей выполнения алгоритма.
Предопределенные процесс		Символ отображает выполнение процесса, который определен в другом месте программы (обычно используется для обозначения вызова функций).
Данные		Используется для обозначения операций ввода-вывода.
Соединитель		Символ обозначает выход в часть схемы и вход из другой части этой схемы. Используется для обрыва линии и продолжения ее в другом месте (пример: разделение блок-схемы, не помещающейся на листе).
Комментарий		Используется для более подробного описания шага, процесса или группы процессов

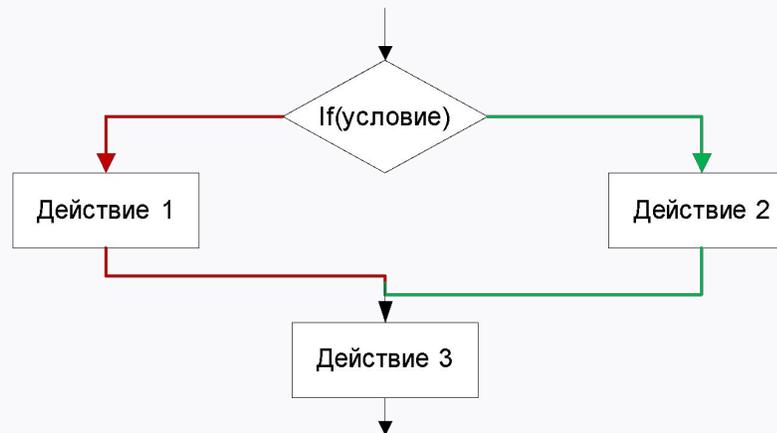


Виды алгоритмов

Линейный

Разветвляющийся

Циклический



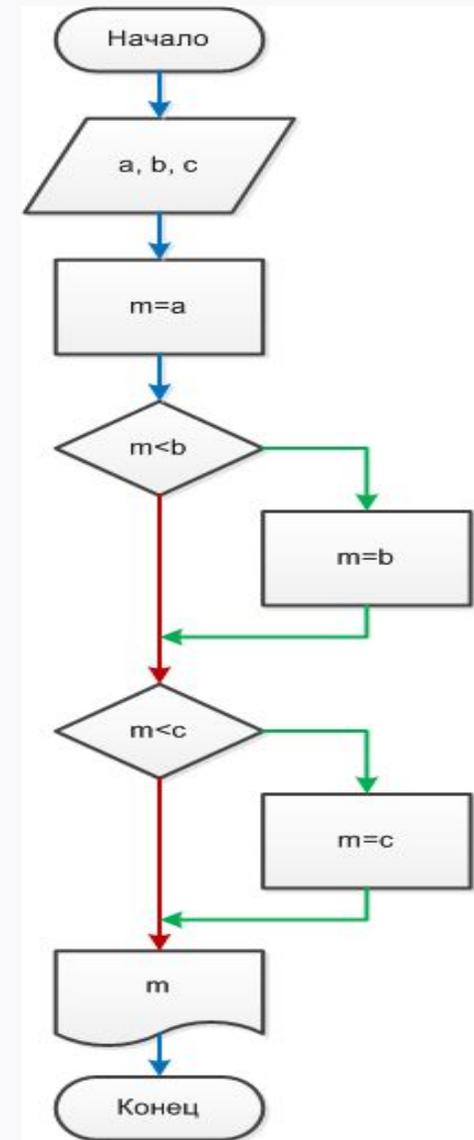
Задача 1. Нахождение максимума из трех чисел

Пусть a , b , c - переменные, которым присваиваются введенные числа, а переменная m в конечном итоге должна будет содержать значение наибольшей переменной. Тогда алгоритм программы сведется к следующим шагам:

Шаг 1. Сначала предположим, что переменная a содержит наибольшее значение. Присвоим его переменной m .

Шаг 2. Если текущее значение m меньше, чем у b , то следует присвоить m значение b . Если это не так, то не изменять значение m .

Шаг 3. Если текущее значение m меньше, чем у c , то присвоить m значение c . Иначе ничего не делать.



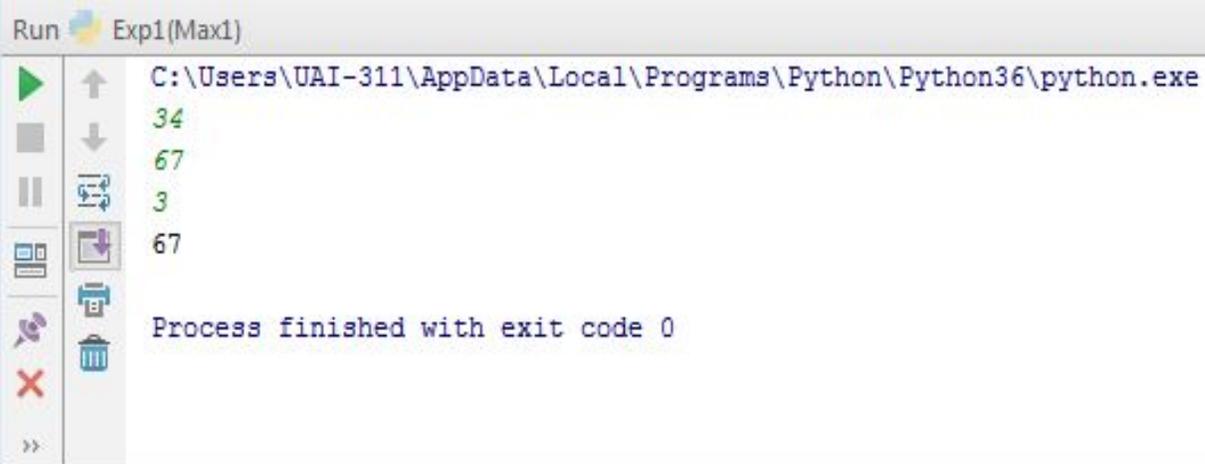
Задача 1. Нахождение максимума из трех чисел

1-й вариант:

```
a = int(input())  
b = int(input())  
c = int(input())
```

```
m = a  
if m < b:  
    m = b  
if m < c:  
    m = c
```

```
print(m)
```



```
Run Exp1(Max1)  
C:\Users\UAI-311\AppData\Local\Programs\Python\Python36\python.exe  
34  
67  
3  
67  
Process finished with exit code 0
```

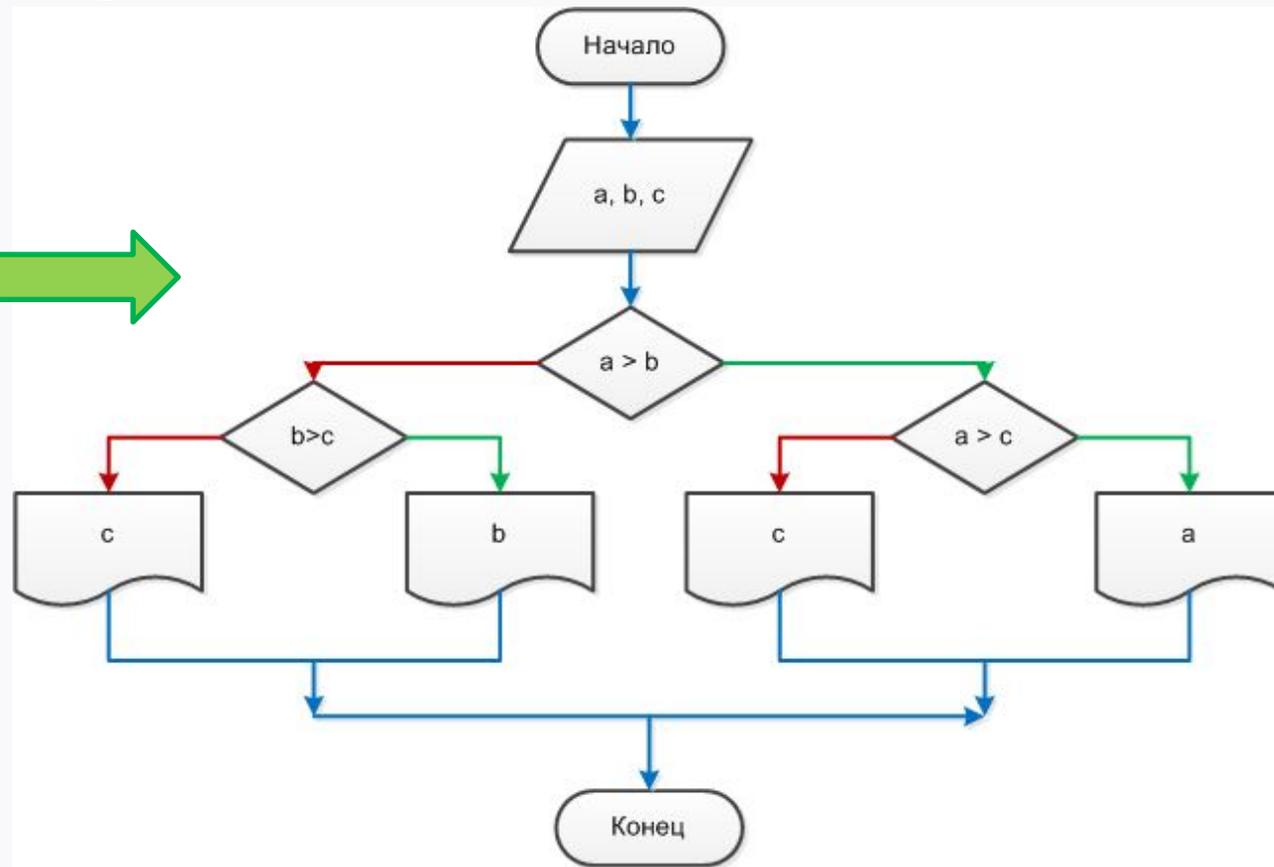


Задача 1. Нахождение максимума из трех чисел

Данную задачу можно решить и без использования четвертой переменной, однако программа будет логически сложнее и будет состоять из следующих шагов:

Шаг 1. Если a больше b , то проверить больше ли a , чем c . Если это так, то максимальное число содержит переменная a . Если a больше b , но при этом меньше c , то максимальным является значение c .

Шаг 2. Иначе (когда a меньше b) сравнить значения b и c . Больше из них и есть максимальное.

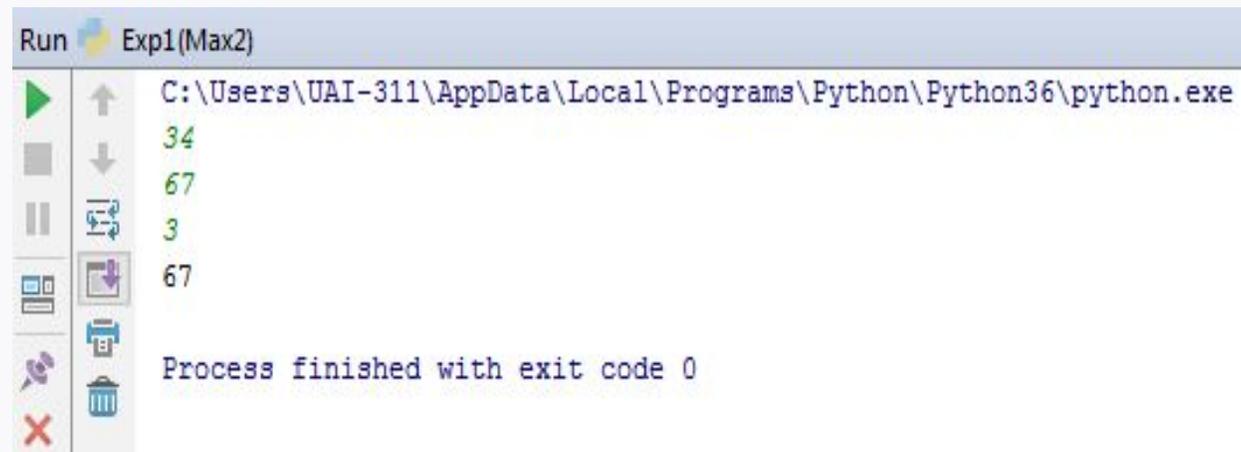


Задача 1. Нахождение максимума из трех чисел

2-й вариант:

```
a = int(input())  
b = int(input())  
c = int(input())
```

```
if a > b:  
    if a > c:  
        print(a)  
    else:  
        print(c)  
else:  
    if b > c:  
        print(b)  
    else:  
        print(c)
```



```
Run Exp1(Max2)  
C:\Users\UAI-311\AppData\Local\Programs\Python\Python36\python.exe  
34  
67  
3  
67  
Process finished with exit code 0
```



Задача 2. Вычислить значение функции $y=f(x)$

Дана следующая функция $y=f(x)$:

$y = 2x - 10$, если $x > 0$

$y = 0$, если $x = 0$

$y = 2 * |x| - 1$, если $x < 0$

Требуется найти значение функции по переданному x .

Алгоритм решения задачи сводится к следующим шагам:

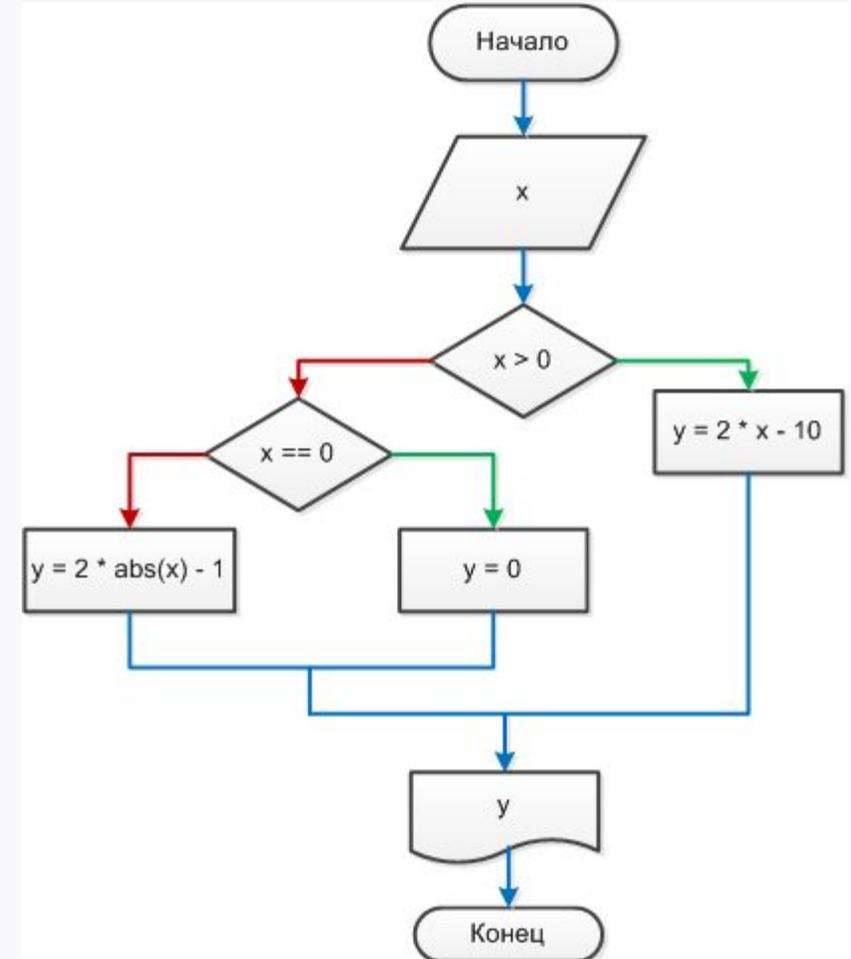
Шаг 1. Получить с клавиатуры значение x .

Шаг 2. Если x больше 0, то вычислить выражение $2*x-10$, результат присвоить переменной y .

Шаг 2.1. Иначе если x равен 0, то присвоить y значение 0.

Шаг 2.2. Иначе присвоить y результат выражения $2*|x|-1$.

Шаг 3. Вывести значение y на экран.

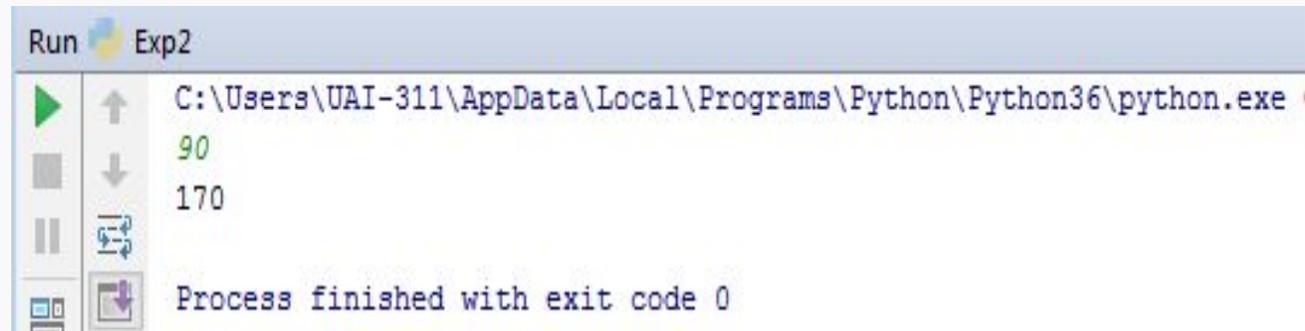


Задача 2. Вычислить значение функции $y=f(x)$

```
x = input()
x = int(x)

if x > 0:
    y = 2 * x - 10
elif x == 0:
    y = 0
else:
    y = 2 * abs(x) - 1

print(y)
```



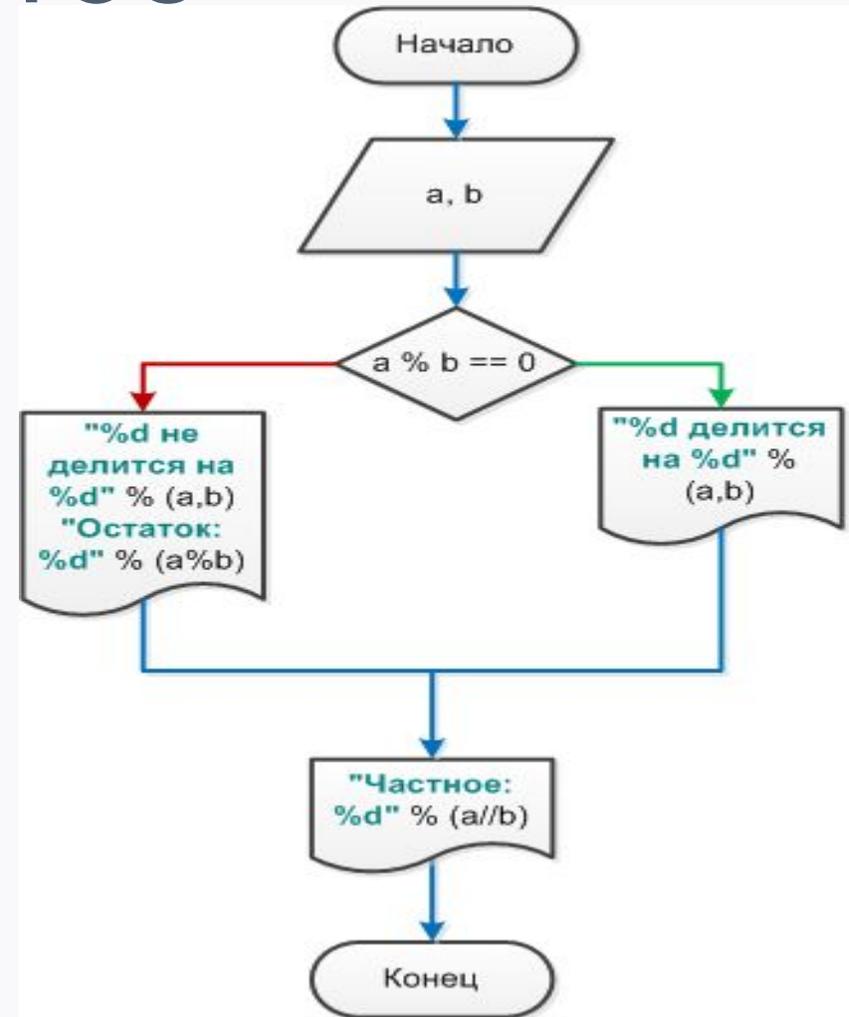
```
Run Exp2
C:\Users\UAI-311\AppData\Local\Programs\Python\Python36\python.exe
90
170
Process finished with exit code 0
```



Задача 3. Проверка делимости одного числа на другое

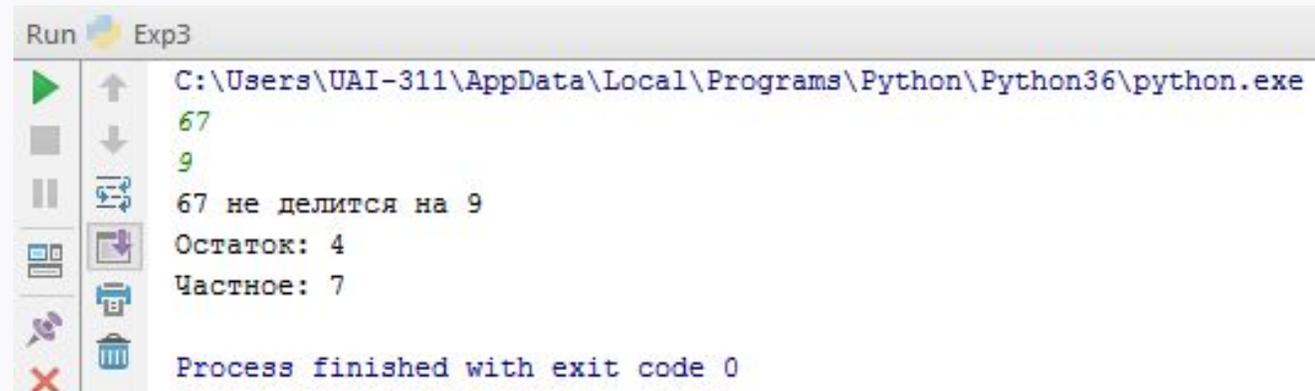
Вводятся два целых числа. Проверить делится ли первое на второе. Вывести на экран сообщение об этом, а также остаток (если он есть) и частное (в любом случае).

Алгоритм решения задачи сводится к следующим шагам:
Шаг 1. Если первое число нацело делится на второе, то вывести сообщение об этом.
Шаг 2. Иначе вывести сообщение о том, что первое число не делится на второе, найти остаток от деления и также вывести его.
Шаг 3. В конце программы найти частное от деления чисел и вывести его.



Задача 3. Проверка делимости одного числа на другое

```
a = int(input())
b = int(input())
if a%b == 0:
    print("%d делится на %d" % (a,b))
else:
    print("%d не делится на %d" % (a,b))
    print("Остаток: %d" % (a%b))
print("Частное: %d" % (a//b))
```



```
Run Exp3
C:\Users\UAI-311\AppData\Local\Programs\Python\Python36\python.exe
67
9
67 не делится на 9
Остаток: 4
Частное: 7
Process finished with exit code 0
```



Задача 4. Перевести байты в килобайты или наоборот

Перевести число, введенное пользователем, в байты или килобайты в зависимости от его выбора.

Алгоритм решения задачи сводится к следующим шагам:

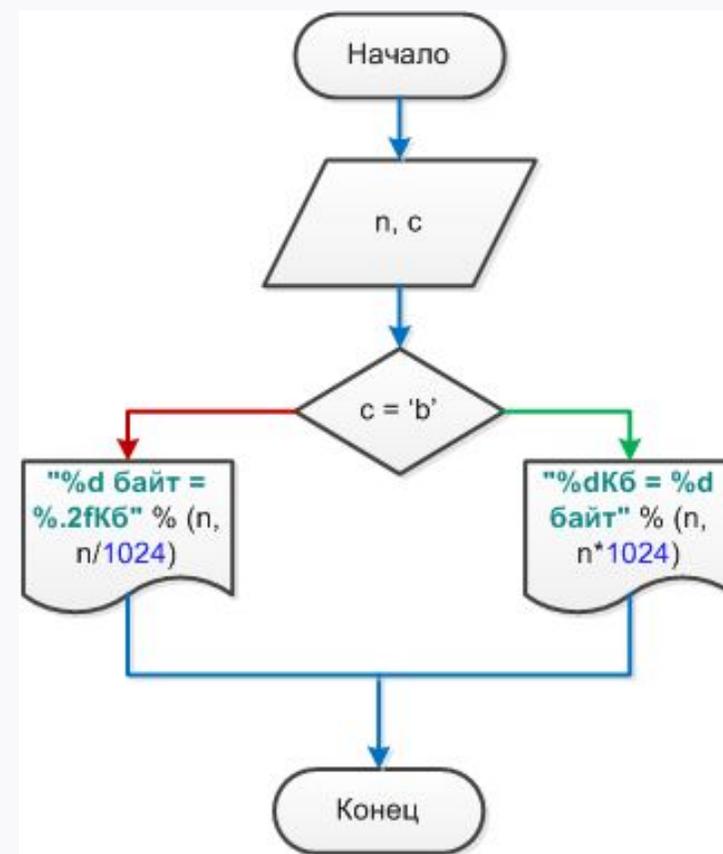
В данной задаче пользователь должен ввести два значения: само число и как-либо указать, во что он хочет его перевести (в байты или килобайты). Пусть выбор способа перевода указывается с помощью одного из двух символов. Например, если пользователь введет букву "b", то число будет переводиться в байты, а если букву "k", то в килобайты. Тогда алгоритм решения задачи будет следующим:

Шаг 1. Ввести число.

Шаг 2. Указать направление перевода.

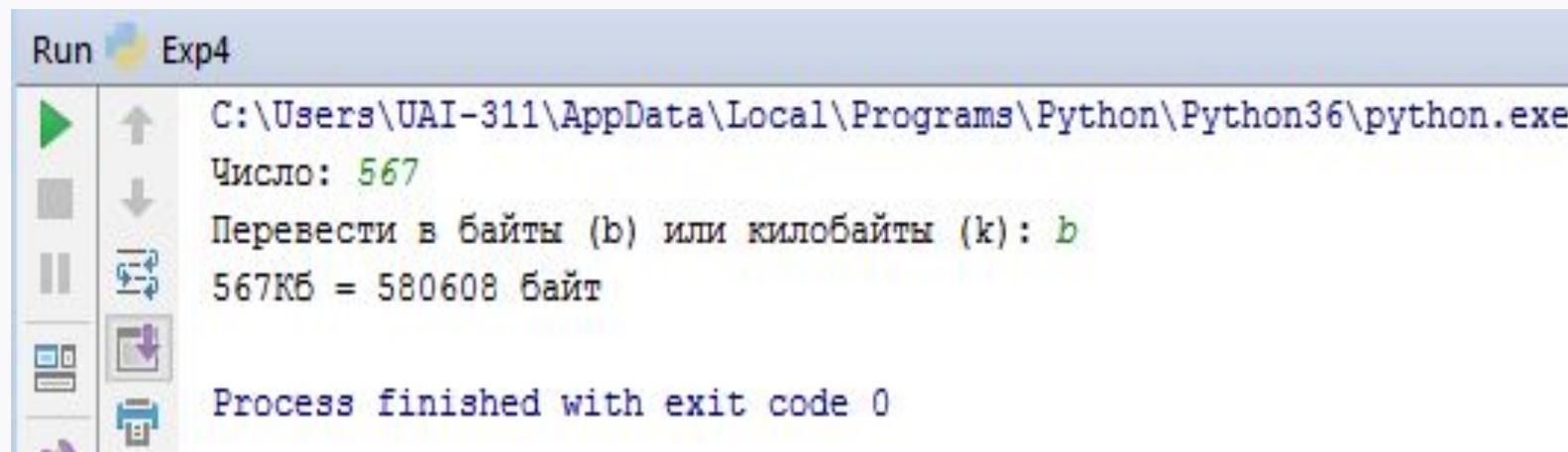
Шаг 3. Если выбран перевод в байты, то надо умножить число на 1024.

Шаг 4. Если выбран перевод в килобайты, то надо разделить число на 1024.



Задача 4. Перевести байты в килобайты или наоборот

```
n = int(input("Число: "))
c = input("Перевести в байты (b) или килобайты (k): ")
if c == 'b':
    print("%dКб = %d байт" % (n, n*1024))
elif c == 'k':
    print("%d байт = %.2fКб" % (n, n/1024))
```



```
Run Exp4
C:\Users\UAI-311\AppData\Local\Programs\Python\Python36\python.exe
Число: 567
Перевести в байты (b) или килобайты (k): b
567Кб = 580608 байт

Process finished with exit code 0
```



Домашнее задание

Для каждого упражнения составить словесное и графическое представление алгоритма и написать программную реализацию.

1. Найти сумму и произведение цифр трехзначного числа, которое вводит пользователь.
2. Выполнить логические побитовые операции "И", "ИЛИ" и др. над числами 5 и 6. Выполнить над числом 5 побитовый сдвиг вправо и влево на два знака. Объяснить полученный результат.
3. По введенным пользователем координатам двух точек вывести уравнение прямой, проходящей через эти точки.
4. Написать программу, которая генерирует в указанных пользователем границах случайное целое число, случайное вещественное число, случайный символ.
5. Для каждого из трех случаев пользователь задает свои границы диапазона. Например, если надо получить случайный символ от 'a' до 'f', то вводятся эти символы. Программа должна вывести на экран любой символ алфавита от 'a' до 'f' включительно.
6. Пользователь вводит две буквы. Определить, на каких местах алфавита они стоят, и сколько между ними находится букв.
7. Пользователь вводит номер буквы в алфавите. Определить, какая это буква.
8. По длинам трех отрезков, введенных пользователем, определить возможность существования треугольника, составленного из этих отрезков. Если такой треугольник существует, то определить, является ли он равносторонним, равнобедренным или равносильным.
9. Определить, является ли год, который ввел пользователь, високосным или не високосным.
10. Вводятся три разных числа. Найти, какое из них является средним (больше одного, но меньше другого).



Дополнительные материалы

1. Ссылка на проекты с практическими примерами.
<https://drive.google.com/open?id=0B2HCiOJRtFFzSWZrOTd5djhaem8>
2. <http://www.intuit.ru/studies/courses/10/320/info>
3. <http://taskcode.ru>
4. Ссылка на среду разработки. <http://www.jetbrains.com/pycharm/>

