

# Lesson 1 Ruby tasks

## Task 1: Обработка строк

### Description:

На вход подается переменная строки.

Написать **однострочную** инструкцию, которая поменяет порядок букв на противоположный и понизит все буквы в регистре кроме первой.

Вернуть значение результата. Оригинальный объект, на который ссылается переменная, должен быть изменен.

Пример:

```
a = "dniMyMdegnahCybuR"
a.object_id # => 12345

# Важно! Сигнатура метода (имя, количество аргументов) должно совпадать с примером
def test(value)
  < ваш код тут >
end

test(a) # => "Rubychangedmymind"
test(a).class # => String
test(a).object_id # => 12345
puts a # => "Rubychangedmymind"
```

## Task 2: Фильтрация чисел

### Description:

Вернуть массив чисел от 10 до 3 не включая 5, причём числа, кратные 3м, вывести в квадрате.

Пример:

```
# Важно! Сигнатура метода (имя, количество аргументов) должно
совпадать с примером
def test
  < ваш код тут >
end

test # => [10, 81, 8, 7, 36, 4, 9]
```

### Task 3: Перестановка цифр

#### **Description:**

Дано число. Вернуть число, где первая цифра стала последней, вторая - предпоследней и тд.

Ноль остается нулем.

Знак минуса также сохраняется.

Примеры:

```
# Важно! Сигнатура метода (имя, количество аргументов) должно совпадать с примером
def test(value)
  < ваш код тут >
end

test(number) # => 321
test(number).class # => Integer

test(-461) # => -164
test(-1) < 0 # => true
```

### Task 4: Сумма цифр

#### **Description:**

Дано целое число. Найти и вернуть сумму его цифр.

Пример:

```
# Важно! Сигнатура метода (имя, количество аргументов) должно совпадать с примером
def test(value)
  < ваш код тут >
end

test(123) == 6 # => true
test(0) == 0 # true
test(-123) == 6 # true
test(19).class == Integer
```

### Task 5: Подсчет вхождения подстрок

#### **Description:**

Дана строка. Необходимо подсчитать количество букв "а" в этой строке (независимо от регистра).

Пример:

```
# Важно! Сигнатура метода (имя, количество аргументов) должно совпадать с примером
def test(string)
  < ваш код тут >
end

test('hello') == 0 # => true
test('arbuz') == 1 # => true
test('AlibaBa') == 3 # => true
```

## Task 6: Палиндром

### **Description:**

Дана строка.

Проверить, является ли она палиндромом (<https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BB%D0%B8%D0%BD%D0%B4%D1%80%D0%BE%D0%BC>)

Регистр значения не играет, знаки пунктуации и отступы – влияют.

```
# Важно! Сигнатура метода (имя, количество аргументов) должно совпадать с примером
def palindrome(value)
  < ваш код тут >
end

palindrome("a") # => true
palindrome("eHohe") # => true
palindrome("rub yybur") # => false
palindrome("ruby ! ybur") # => true
```

## Task 7: Поиск значения в хэше

### **Description:**

Дан хэш:

```
shop = {
  milk: 10,
  bread: 8,
  cornflakes: 12,
  ice_cream: 15,
  pie: 20
}
```

Ответить на вопрос: если ли в магазине какой-либо продукт с заданой ценой?

Пример:

```
# Важно! Сигнатура метода (имя, количество аргументов) должно совпадать с примером
def test(value)
  shop = {
    milk: 10,
    bread: 8,
    cornflakes: 12,
    ice_cream: 15,
    pie: 20
  }

  < ваш код тут >
end

test(15) # => true
test(100) # => false
```

### **Task 8: Поиск в массивах**

#### **Description:**

Прибавить 100 к максимальному элементу заданного массива и вернуть его.

Пример:

```
# Важно! Сигнатура метода (имя, количество аргументов) должно совпадать с примером
def test(array)
  < ваш код тут >
end

test([1, 6, 1, 8, 2, -1, 3, 5]) # => [1, 6, 1, 108, 2, -1, 3, 5]
test([1, 6, 1, 8, 2, -1, 3, 5]).class == Array # => true
test([]) # => []
test([1]) # => [101]
test([1, 1]) # => [101, 101]
```

## Task 9: Сортировка массивов

### **Description:**

Вывести отсортированный по убыванию массив из уникальных элементов многомерного массива.

Пример:

```
# Важно! Сигнатура метода (имя, количество аргументов) должно совпадать с примером
def test(array)
  < ваш код тут >
end

ar = [7, 3, [4, 5, 1], 1, 9, [2, 8, 1]]

test(ar) == [9, 8, 7, 5, 4, 3, 2, 1] # => true
```

## Task 10: Последовательность чисел Фибоначчи

### **Description:**

Написать код, который выводит массив из n-первых элементов последовательности чисел [Фибоначчи](#)

Решение должно работать для любого n.

Пример:

```
$ ./script.rb 7
=> 0 1 1 2 3 5 8

# Важно! Сигнатура метода (имя, количество аргументов) должно совпадать с примером
def fib(n)
  < ваш код тут >
end

fib(7) == [0, 1, 1, 2, 3, 5, 8] # => true
fib(0) == [] # => true
fib(1) == [0] # => true
fib(3) == [0, 1, 1] # => true
```