



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т

по лабораторной работе № _ 5

Название: Основы Ruby.

Дисциплина: Языки интернет программирования

Студент

ИУ6-31Б

(Группа)

(Подпись, дата)

Т.Е. Старжевский

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Д. В. Малахов

(И.О. Фамилия)

Москва, 2022

Текст задания:

Часть 1

Вычислить: $y = \cos(x + 3.1 \cdot z) / \operatorname{tg}(x/r)$

Часть 2

Дан одномерный массив числовых значений, насчитывающий N элементов. Определить, образуют ли элементы массива, расположенные перед первым отрицательным элементом, возрастающую последовательность.

Часть 3

Дана последовательность строк. Каждая строка состоит из слов, разделенных пробелами. Написать программу, обеспечивающую ввод строк и их корректировку. Корректировка заключается в следующем. Если слово содержит символы, отличные от букв латинского алфавита и цифр, то удалить его. Если слово состоит из букв латинского алфавита и цифр и начинается с цифры, заменить эту цифру символом «_» подчеркивание.

Часть 1

Текст программ:

main.rb:

```
# tg(x/r)
class Solution
  include Math

  def self.calculate(x, z, r)
    raise 'ZeroDivisionError' if x.zero? || Math.tan(x / r).zero?

    (Math.cos(x + 3.1 * z) / Math.tan(x / r)).floor(2)
  rescue StandardError => e
    e.message
  end
end
```

user.rb:

```
require_relative 'main'

print "y = cos(x + 3.1 * z) / tg(x / r)\n\tInput x: "
x = gets.to_f
print "\tInput z: "
z = gets.to_f
print "\tInput r: "
r = gets.to_f

puts "x = #{x} #{x.class}\tz = #{z} #{z.class}\tr = #{r} #{r.class}"

result = Solution.calculate x, z, r
puts "Result = #{result}"
```


test.rb:

```
require 'minitest/autorun'
require_relative 'main'

# Test class
class MyTest < Minitest::Test
  def test_error
    result = (Solution.calculate 1, 5, 2)
    puts "result == #{result}"
    assert (Solution.calculate 1, 5, 2) == 'ZeroDivisionError'
  end

  def test_float_values
    assert (Solution.calculate 1.0, 5.0, 2.0).truncate == -1
  end

  def test_values
    result = (Solution.calculate 1.0, 5.0, 2.0).truncate
    puts "result == #{result}"
    assert (Solution.calculate 6.0, 7.0, 1.0).truncate == 2
  end
end
```

Результаты работы и тестов:

```

• timofey@timofey-ASUS:~/Projects/BMSTU_WEB/Labs/lab5/2022_part_1$ ruby user.rb
y = cos(x + 3.1 * z) / tg(x / r)
    Input x: 15
    Input z: 15
    Input r: 10
x = 15.0 Float  z = 15.0 Float  r = 10.0 Float
Result = 0.01
• timofey@timofey-ASUS:~/Projects/BMSTU_WEB/Labs/lab5/2022_part_1$ rspec test.rb
No examples found.

```

```

Finished in 0.00031 seconds (files took 0.07833 seconds to load)
0 examples, 0 failures

```

```

Run options: --seed 3367

```

```

# Running:

```

```

result == ZeroDivisionError
..result == -1
.

```

```

Finished in 0.004572s, 656.2189 runs/s, 656.2189 assertions/s.
3 runs, 3 assertions, 0 failures, 0 errors, 0 skips

```

```

• timofey@timofey-ASUS:~/Projects/BMSTU_WEB/Labs/lab5/2022_part_1$ 
main.rb:1:1: C: [Correctable] Style/FrozenStringLiteralComment: Missing frozen string literal
comment.
# tg(x/r)
^
main.rb:5:22: C: Naming/MethodParameterName: Method parameter must be at least 3 characters lo
ng.
  def self.calculate(x, z, r)
                    ^
main.rb:5:25: C: Naming/MethodParameterName: Method parameter must be at least 3 characters lo
ng.
  def self.calculate(x, z, r)
                    ^
main.rb:5:28: C: Naming/MethodParameterName: Method parameter must be at least 3 characters lo
ng.
  def self.calculate(x, z, r)
                    ^

```

Часть 2

Текст программ:

main2.rb:

```

# This class is responsible for checking is the elements before first negative sorted from min to max.

```

```

class Solution
  def self.posled?(arr)
    new = arr.take_while { |el| el >= 0 }
    new.empty? || arr == new ? false : new == new.sort
  end
end

```

user.rb:

```

require_relative 'main'

print 'Input array: '
solution = Solution.new(gets.split.map(&:to_f))

if solution.posled?
  puts "Yes, posled: #{solution.arr_new}"
else

```

```
puts 'No'  
end
```

test2.rb:

```
require_relative 'main2'
```

```
RSpec.describe Solution do
```

```
  describe '#Solution' do
```

```
    it 'should return true' do
```

```
      expect(Solution.posled?([1, 2, 3])).to eq(false)
```

```
    end
```

```
    it 'should return true if contains posled before negative el' do
```

```
      uncorrect = [1, 2, 3, -1]
```

```
      Random.rand(10).times { uncorrect.push(Random.rand(-10..9)) }
```

```
      expect(Solution.posled?(uncorrect)).to eq(true)
```

```
    end
```

```
    it 'should return false if there is no posled before first neg el' do
```

```
      uncorrect = [1, 3, 2, -1]
```

```
      Random.rand(10).times { uncorrect.push(Random.rand(-10..9)) }
```

```
      expect(Solution.posled?(uncorrect)).to eq(false)
```

```
    end
```

```
    it 'should return false if first el negative' do
```

```
      uncorrect = [-1, 1, 2, 3]
```

```
      Random.rand(10).times { uncorrect.push(Random.rand(-10..9)) }
```

```
      expect(Solution.posled?(uncorrect)).to eq(false)
```

```
    end
```

```
    it 'should return false if there is no negative elements' do
```

```
      uncorrect = [1, 2, 3]
```

```
      Random.rand(10).times { uncorrect.push(Random.rand(-10..9)) }
```

```
      expect(Solution.posled?(uncorrect)).to eq(false)
```

```
    end
```

```
  end
```

```
end
```

Результаты работы и тестов:

```

● timofey@timofey-ASUS:~/Projects/BMSTU_WEB/Labs/lab5/2022_part_2$ ruby user.rb
Input array: 8 16 -4 9 46 88 99 -5 0
Yes, posled: [8.0, 16.0]
● timofey@timofey-ASUS:~/Projects/BMSTU_WEB/Labs/lab5/2022_part_2$ rspec test2.rb
.....

Finished in 0.00235 seconds (files took 0.07079 seconds to load)
5 examples, 0 failures

○ timofey@timofey-ASUS:~/Projects/BMSTU_WEB/Labs/lab5/2022_part_2$ █

```

```

main.rb:1:1: C: [Correctable] Style/FrozenStringLiteralComment: Missing frozen string literal
comment.
# This class is responsible for checking is the elements before first negative sorted from min
to max.
^
main.rb:5:3: C: Metrics/MethodLength: Method has too many lines. [11/10]
def initialize(arr) ...
~~~~~

```

Часть 3

Текст программ:

main2.rb:

```

class Solution2
  attr_accessor :counter

  def initialize(strings)
    @strings = strings
    @reg_exp = /^[ёЁа-яА-Я0-9a-zA-Z]*$/
    @counter = 0
  end

  def make_correct_with_count(strings = @strings)
    strings.map! do |string|
      selected = string.split.select do |el|
        @counter += 1 unless el.match(@reg_exp)
      end
      el.match(@reg_exp)
    end
    selected.map! do |word|
      @counter += 1 if word.match(/^\d/)
      word.sub(/^\d/, '_')
    end
    selected.join(' ')
  end

  strings.reject(&:empty?)
end

def make_correct(strings = @strings)
  strings.map! do |string|
    string.split.select { |el| el.match(@reg_exp) }.map! { |word| word.sub(/^\d/, '_') }.join(' ')
  end
  strings.reject(&:empty?)
end

```

```
end
end
```

user.rb:

```
require_relative 'main'

print "Input strings, for end type '#': "
str = ""
strings = []
while str != '#'
  strings.push(str)
  str = gets.chomp
end
solution = Solution.new(strings)
puts '===== Output ====='
puts solution.make_correct
```

test2.rb:

```
require_relative 'main'
require_relative 'main2'

# Class for test Solution from main
RSpec.describe Solution do
  describe '#main' do
    it 'should return not changed strings' do
      test_strings = ['Ты красивая как осень хочу тебя целовать', 'В твоей голове вопросы В моей просто пустота']
      correct_strings = Solution.new(test_strings).make_correct
      expect(correct_strings).to eq(test_strings)
    end

    it 'should return with deleting strings contains only spec symbols' do
      test_strings = ['Ты красивая как осень хочу тебя целовать', 'В твоей голове вопросы В моей просто пустота']
      correct_strings = Solution.new(test_strings + ['&hasd', '*dasf', 'asd']).make_correct
      expect(correct_strings).to eq(test_strings)
    end

    it 'should returns true for auto generating strings' do
      # ind = 0
      5000.times do
        symbols = "~`!@#$%^&*()_-=\\\"№,:;?\\|/[]{}'.,/\"
        correct_strings = ['Ты красивая как осень хочу тебя целовать', 'В твоей голове вопросы В моей просто пустота',
          'Ты же знаешь В любви просто не бывает никогда', 'Потому я молча взял тебя просто поцеловал',
          'Ты красивая как осень ну а я простой пацан', 'У меня Р00 рублей McDonalds тоже ресторан',
          'Ты же знаешь пацаны мы не берём слова назад', 'Так что наши поцелуи с тобой это навсегда',
          'Обнимай меня каждый день', 'Засыпай на моих руках', 'Ты же знаешь как важно мне',
          'Знать у нас всё не просто так', 'Обнимай меня каждый день', 'Засыпай на моих руках',
          'Ты же знаешь как важно мне', 'Знать у нас всё не просто так']
        uncorrect_strings = []
        correct_strings.each do |string|
          unc_string = string.split.inject("") do |acc, word|
            if Random.rand(2) == 1
```



```

word << ''
Random.rand(2..8).times { word << symbols[Random.rand(symbols.length)] }
end
acc << word << ''
end
uncorrect_strings.push(unc_string[0...-1])
end

expect(Solution.new(uncorrect_strings).make_correct).to eq(correct_strings)
# ind += 1
# if (ind == 5)
# expect(Solution.new(uncorrect_strings).make_correct).to eq(correct_strings + ["test"])
# end
end
end
end
end

# Class for test Solution from main2
RSpec.describe Solution2 do
  describe '#main2' do
    it 'should return not changed strings' do
      test_strings = ['Ты красивая как осень хочу тебя целовать', 'В твоей голове вопросы В моей просто пустота']
      correct_strings = Solution2.new(test_strings).make_correct
      expect(correct_strings).to eq(test_strings)
    end

    it 'should return with deleting strings contains only spec symbols' do
      test_strings = ['Ты красивая как осень хочу тебя целовать', 'В твоей голове вопросы В моей просто пустота']
      correct_strings = Solution2.new(test_strings + ['&hasd', '*dasf', 'asd']).make_correct
      expect(correct_strings).to eq(test_strings)
    end

    it 'should returns true for auto generating strings' do
      # ind = 0
      5000.times do
        symbols = "~`!@#$%^&*()_-=\\\"№,:;?\\|/[]{}'.,/"
        correct_strings = ['Ты красивая как осень хочу тебя целовать', 'В твоей голове вопросы В моей просто пустота',
          'Ты же знаешь В любви просто не бывает никогда', 'Потому я молча взял тебя просто поцеловал',
          'Ты красивая как осень ну а я простой пацан', 'У меня Р00 рублей McDonalds тоже ресторан',
          'Ты же знаешь пацаны мы не берём слова назад', 'Так что наши поцелуи с тобой это навсегда',
          'Обнимай меня каждый день', 'Засыпай на моих руках', 'Ты же знаешь как важно мне',
          'Знать у нас всё не просто так', 'Обнимай меня каждый день', 'Засыпай на моих руках',
          'Ты же знаешь как важно мне', 'Знать у нас всё не просто так']
        uncorrect_strings = []
        correct_strings.each do |string|
          unc_string = string.split.inject("") do |acc, word|
            if Random.rand(2) == 1
              word << ''
              Random.rand(2..8).times { word << symbols[Random.rand(symbols.length)] }
            end
            acc << word << ''
          end
          uncorrect_strings.push(unc_string[0...-1])
        end
      end
    end
  end
end

```

```

end
uncorrect_strings.push(unc_string[0...-1])
end

expect(Solution2.new(uncorrect_strings).make_correct).to eq(correct_strings)
end
end
end
end

```

Результаты работы и тестов:

```

• timofey@timofey-ASUS:~/Projects/BMSTU_WEB/Labs/lab5/2022_part_3$ ruby user.rb
Input strings, for end type '#': Ты красивая как осень хочу тебя целовать
Ты красивая как осень ну а я простой пацан
Ты красивая как осень ну а я простой пацан
У меня Р00 рублей McDonalds тоже рестор 5абра
9убери 8тут 8тоже а_тут_не
#
===== Output =====
Ты красивая как осень хочу тебя целовать
Ты красивая как осень ну а я простой пацан
Ты красивая как осень ну а я простой пацан
У меня Р00 рублей McDonalds тоже рестор _абра
_убери _тут _тоже
• timofey@timofey-ASUS:~/Projects/BMSTU_WEB/Labs/lab5/2022_part_3$ rspec test2.rb
["Ты красивая как осень хочу тебя целовать", "В твоей _удалиЦифру голове вопросы В моей просто
пустота"]
0
.....

Finished in 3.33 seconds (files took 0.08877 seconds to load)
6 examples, 0 failures

○ timofey@timofey-ASUS:~/Projects/BMSTU_WEB/Labs/lab5/2022_part_3$

```

```

main2.rb:1:1: C: [Correctable] Style/FrozenStringLiteralComment: Missing frozen string literal
comment.
# Дана последовательность строк. Каждая строка состоит из слов, разделенных
^
main2.rb:20:3: C: Metrics/MethodLength: Method has too many lines. [12/10]
def make_correct_with_count(strings = @strings) ...
~~~~~

```

Вывод: Научился базовой с языком Ruby.