

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 09.03.03 «Прикладная информатика»

ОТЧЕТ

по лабораторной работе № 9

Название: Создание простейших веб-приложений Ruby on Rails.

<u>AJAX</u>

Дисциплина: Языки Интернет-программирования

 Студент
 ИУ6-34Б
 20.11.2022
 С. А. Рахманов

 (Группа)
 (Подпись, дата)
 (И.О. Фамилия)

Преподаватель Д. В. Малахов

(Подпись, дата) (И.О. Фамилия)

углубление Цель работы: теоретических сведений принциработы асинхронного веб-интерфейса пах получение практиченавыков создания веб-приложения использованием средств СКИХ C Ruby on Rails и технологии AJAX.

Задание:

Все консольные приложения Ruby следует реализовывать в виде трех отдельных файлов:

- 1. основная программа;
- 2. программа для взаимодействия с пользователем через консоль;
- 3. программа для автоматического тестирования на основе MiniTest::Unit или RSpec. Везде, где это возможно, данные для проверки должны формироваться автоматически по правилам, указанным в задании.

При реализации программ везде, где это возможно, следует избегать использования циклов for, do, while. Вместо них используйте методы из примеси Enumerable.

Все тексты программ должны быть проверены на соответствие стилю программирования Ruby при помощи rubocop.ru или reek.

Разработать веб-приложение, имеющее HTML-страницу с формой ввода данных и HTML-страницу для представления результатов. Результат расчёта должен быть представлен в форме таблицы, оформленной с помощью элемента table или отдельными ячейками div и имеющей не менее двух колонок. Если по условию задания результат может быть представлен только в виде одной строки таблицы, необходимо реализовать вывод промежуточных результатов расчёта в качестве дополнительных строк. В этом случае первой колонкой таблицы будет порядковый номер итерации.

Под вводом с клавиатуры в тексте заданий следует понимать ввод в поле ввода данных формы на HTML-странице.

Текст задания:

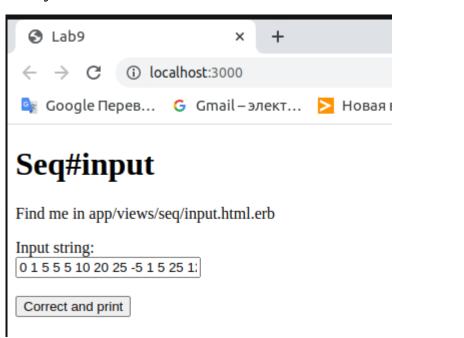
Написать программу, которая вводит цепочку целых чисел (количество чисел не менее 10) и определяет наиболее длинную монотонно возрастающую их последовательность. Вывести на печать введенную цепочку, все найденные последовательности и наиболее длинную из них. При программировании использовать функцию.

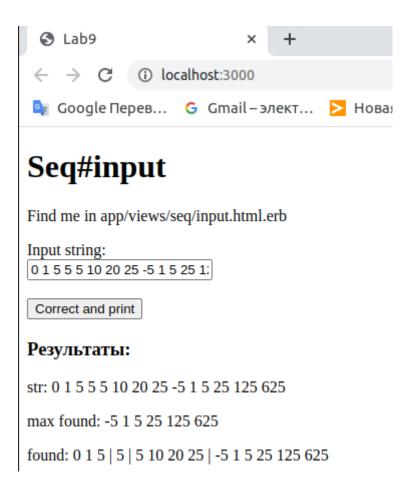
Тексты измененных файлов

```
<u>seq_controller.</u>rb
      def correct(arr)
        found = []
        tmp found = []
        arr.take(arr.size - 1).each index do |i|
          tmp found << arr[i]</pre>
          unless arr[i] < arr[i + 1]
            found << tmp_found</pre>
            tmp found = []
          end
        end
        tmp found << arr[-1]</pre>
        found << tmp found
        sizes = found.map(&:size)
        @max found = found[sizes.index(sizes.max)].join(' ')
        @found = found.map \{ |x| x.join(' ') \}.join(' | ')
      end
      # Top-level class
      class SeqController < ApplicationController</pre>
        def input; end
        def view
          @str = !params[:str] || params[:str] == '' ? 'Error' : params[:str]
          @max found = @str
          @found = @str
          arr = @str&.split&.map(&:to i)
          arr && !arr&.empty? ? correct(arr) : nil
          respond to do |format|
            format.html
            format.js
          end
        end
      end
input.html.erb
      <h1>Sea#input</h1>
      Find me in app/views/seq/input.html.erb
      <%= form_with url: '/seq/view.js', method: :get, local: false do |form| %>
           <%= form.label("Input string:") %> <br/>
          <%= form.text field(:str, value: "0 1 5 5 5 10 20 25 -5 1 5 25 125 625")</pre>
      %> <br/><br/>
          <%= form.submit("Correct and print")%> <br/>
      <% end %>
      <div id="result"></div>
view.js.erb
      document.getElementById('result').innerHTML = `
          <h3>Результаты:</h3>
           <%== '<p>HET ПОСЛЕДОВАТЕЛЬНОСТЕЙ' if !@found %>
           <%== alert if @str&.match(/[\s\d]/).nil? %>
           str: <%= @str %>
           max found: <%= @max_found %>
           found: <%= @found %>
```

```
routes.rb
      Rails.application.routes.draw do
        get 'seq/input'
        get 'seq/view'
        root 'seq#input'
        # Define your application routes per the DSL in
      https://guides.rubyonrails.org/routing.html
        # Defines the root path route ("/")
        # root "articles#index"
      end
<u>seq_controller_test.rb</u>
      # frozen_string_literal: true
      require 'test helper'
      class SeqControllerTest < ActionDispatch::IntegrationTest</pre>
        test 'should get input' do
          get seq_input_url
          assert_response :success
        end
        test 'should get view' do
          get seq_view_url
          assert response :success
        end
      end
```

Результаты выполнения:





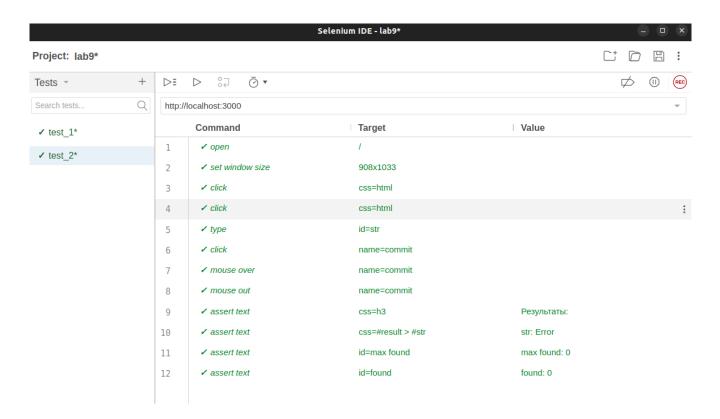
Результаты проверки анализатором rubocop:

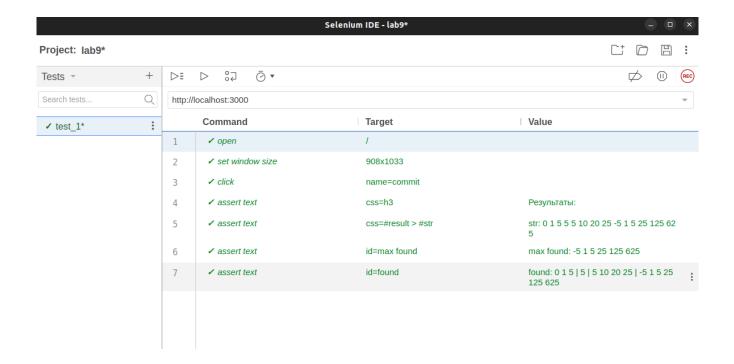
```
Tepминал

S3r6anita@s3r6anita-Modern-14-B10MW:~/LoIP/labs/lab9/lab9/app$ rubocop controlle rs
Inspecting 2 files

1 files inspected, no offenses detected s3r6anita@s3r6anita-Modern-14-B10MW:~/LoIP/labs/lab9/lab9/app$
```

Тестирование с помощью Selenium IDE:





Вывод: в результате выполнения лабораторной работы были получены практические навыки создания асинхронного веб-интерфейса, интеграционного тестирования с помощью среды Sleneium IDE.