

(МГТУ им. Н.Э. Баумана)

КАФЕДРА Компьютерные системы и сети(ИУ6)

## по лабораторной работе № 9\_\_\_\_

Дисциплина: Языки интернет-программирования

**Название лабораторной работы:** Ruby on Rails. AJAX

Преподаватель \_\_\_\_\_ (Подпись, дата) \_\_\_\_\_ (И.О. Фамилия)

Москва, 2020

# Задание

При помощи Javascript модифицировать код ЛР 8 таким образом, чтобы для отображения результатов вычисления браузер не выполнял полную перезагрузку страницы.

1. Сформировать тесты для проверки работы программы при помощи Katalon Recorder / Selenium Webdriver.

## Код программы input.html.erb:

```
<h2>Input</h2>
<hr>
<%= form_tag("output.json", :method => "get", remote: true, id: 'input_form') do %>
  <%= text_field_tag(:sequence) %>
  <%= submit_tag("Определить наиболее длинную монотонно возрастающую последовательность") %>
<% end %>

<div id="result">
  <!-- поле для асинхронного вывода результатов вычислений -->
</div>
```

## result.html.erb

```
<h2>Output</h2>
<div id="result">
</div>
```

## io\_controller.rb:

```
# frozen_string_literal: true

# output
class IoController < ApplicationController
  def output
    sequence = params[:sequence]
    @result = "hello"

    if sequence == ""
      @result = 'you did not write anything:'
    elsif sequence.to_i.zero?
      @result = 'please, enter a number'
    else
      res = find(sequence)
      if res.length.zero?
        @result = 'There are no rising subsequences'
      else
        @result = res
      end
    end

    respond_to do |format|
      format.html
      format.json do
        render json: {type: @result.class.to_s, value: @result}
      end
    end
  end

  def find(sequence)
    rising_subsequences = []
    subsequence = sequence[0]
```

```

count = 1
n = 0
(1..sequence.length).each do |i|
  if sequence[i].to_i > sequence[i - 1].to_i
    count += 1
    subsequence += sequence[i]
  else
    if count > 1
      rising_subsequences[n] = subsequence
      n += 1
    end
    count = 1
    subsequence = sequence[i]
  end
end
rising_subsequences
end
end

```

### io\_controller.js:

```

function show_result(data)
{
  var result = document.getElementById("result");
  insertToHTML = "<table>";
  if ( data.type == "Array" )
  {
    arr = data.value;
    max = arr[0];
    for ( i = 0 ; i < arr.length - 1 ; i++ )
    {
      insertToHTML += "<tr><td>" + (i+1) + "</td><td>" +
        arr[i] + "</td></tr>";
      if (arr[i].length > max.length) max = arr[i];
    }
    insertToHTML += "<tr><td>" + arr.length + "</td><td>" +
      data.value[arr.length-1] + "</td></tr>";
    insertToHTML = "<hr/>" + "maximum length rising subsequence: " + max + insertToHTML;
  } else
  {
    insertToHTML += data.value;
  }
  insertToHTML += "</table>";
  result.innerHTML = "<hr/>" + insertToHTML;
}

handleAjaxSuccess = function (event) {
  [data, status, xhr] = event.detail;
  show_result(data);
}

document.addEventListener('DOMContentLoaded', function () {
  document.querySelector("#input_form").addEventListener(
    'ajax:success', handleAjaxSuccess)
})

```

### output\_controller\_test.rb:

```

require 'test_helper'

class OutputControllerTest < ActionDispatch::IntegrationTest
  test "get output" do
    get result_path
    assert_response :success
  end
end

```

```

end

test "should get 'no rising' for test with 11111" do
  get result_path, params: {sequence: 11111}
  assert_equal assigns[:result], "There are no rising subsequences"
end

test "should get an error for NaN" do
  get result_path, params: {sequence: "qwerty"}
  assert_equal assigns[:result], "please, enter a number"
end

test "should get an error for empty input" do
  get result_path, params: {sequence: ""}
  assert_equal assigns[:result], "you did not write anything:("
end
end

```

## Результаты выполнения:

### Основная программа:

Вводим пустую строку:

#### Input

---

	Определить наиболее длинную монотонно возрастающую последовательность
--	---

---

you did not write anything:(

Вводим не числовую строку:

#### Input

---

sdffsdff	Определить наиболее длинную монотонно возрастающую последовательность
----------	---

---

please, enter a number

Вводим последовательность без возрастающих подпоследовательностей:

#### Input

---

11111	Определить наиболее длинную монотонно возрастающую последовательность
-------	---

---

There are no rising subsequences

Хорошие данные:

## Input

123451209346

Определить наиболее длинную монотонно возрастающую последовательность

maximum length rising subsequence: 12345

1 12345

2 12

3 09

4 346

Тесты:

```
labzunova@labzunova:~/Рабочий стол/labs BMSTU/LR9/test$ rake
(in /home/labzunova/Рабочий стол/labs BMSTU/LR9)
Run options: --seed 63668

# Running:

....

Finished in 0.148314s, 26.9699 runs/s, 26.9699 assertions/s.
4 runs, 4 assertions, 0 failures, 0 errors, 0 skips
```

## Проверка rubocop и reek:

Rubocop:

```
labzunova@labzunova:~/Рабочий стол/3-semesterBMSTU/LR9/app/controllers$ rubocop
Inspecting 2 files
..

2 files inspected, no offenses detected
```

Reek:

```
labzunova@labzunova:~/Рабочий стол/3-semesterBMSTU/LR9/app/controllers$ reek
Inspecting 2 file(s):
..

0 total warnings
labzunova@labzunova:~/Рабочий стол/3-semesterBMSTU/LR9/app/controllers$
```