

(МГТУ им. Н.Э. Баумана)

КАФЕДРА Компьютерные системы и сети(ИУ6)

по лабораторной работе № 8\_\_

Дисциплина: Языки интернет-программирования

Название лабораторной работы: Ruby on Rails

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

Москва, 2020

## Задание

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.

### Код программыindex.html.erb:

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

### output.html.erb

```
<h2>Output</h2>
<hr>
<%= @output %>
<hr>

<% if @subsequences.is_a?(Array) %>
  <%= @subsequences %>
<% end %>

<% if @error == 0 %>
  <table>
    <% @subsequences.each_with_index do |sequence, index| %>
      <tr>
        <td><%= index %>:</td>
        <td><%= sequence %></td>
      </tr>
    <% end %>
  </table>
```

```
<% end %>
</table>
```

```
<hr>
```

```
<%= @max %>
<% end %>
```

```
<%= link_to 'Назад', :back %>
```

## output\_controller.rb:

```
# frozen_string_literal: true
```

```
# output
class OutputController < ApplicationController
  def output
    sequence = params[:sequence]
    @error = 1
    if sequence == ""
      @output = 'you did not write anything:('
    elsif sequence.to_i.zero?
      @output = 'please, enter a number'
    else
      @output = "Your initial sequence: #{sequence}"
      @error = 0
      res = find(sequence)
      if res[0].length.zero?
        @subsequences = 'There are no rising subsequences'
        @error = 1
      else
        @subsequences = res[0]
        @max = "maximum length rising subsequence: #{res[1]}"
      end
    end
  end
end
```

```
def find(sequence)
  rising_subsequences = []
  subsequence = sequence[0]
  max = ""
  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
        max = subsequence if subsequence.length > max.length
        n += 1
      end
      count = 1
      subsequence = sequence[i]
    end
  end
  [rising_subsequences, max]
end
```

## output\_controller\_test.rb:

```
require 'test_helper'

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

  test "should get 123 for test with 12323180" do
    get '/output', params: {sequence: 12323180}
    assert_equal assigns[:max], "maximum length rising subsequence: 123"
  end

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

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

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

## Результаты выполнения:

### Основная программа:

Вводим пустую строку:

### Output

---

you did not write anything:(

---

[Назад](#)

Вводим не числовую строку:

### Output

---

please, enter a number

---

[Назад](#)

Вводим последовательность без возрастающих

### Output

---

Your initial sequence: 111111

---

There are no rising subsequences [Назад](#)

## Output

---

Your initial sequence: 123451209346

---

0: 12345

1: 12

2: 09

3: 346

---

maximum length rising subsequence: 12345 [Назад](#)

Хорошие данные: Тесты:

Тесты:

```
labzunova@labzunova:~/Рабочий стол/3-semesterBMSTU/LR8$ rails test
Run options: --seed 19788

# Running:

.....

Finished in 0.100321s, 49.8399 runs/s, 49.8399 assertions/s.
5 runs, 5 assertions, 0 failures, 0 errors, 0 skips
```

## Проверка rubocop и reek:

Rubocop:

```
labzunova@labzunova:~/Рабочий стол/3-semesterBMSTU/LR8/app/controllers$ rubocop
Inspecting 2 files
..

2 files inspected, no offenses detected
labzunova@labzunova:~/Рабочий стол/3-semesterBMSTU/LR8/app/controllers$
```

Reek:

Reek:

```
labzunova@labzunova:~/Рабочий стол/3-semesterBMSTU/LR8/app/controllers$ reek
Inspecting 2 file(s):
..

0 total warnings
labzunova@labzunova:~/Рабочий стол/3-semesterBMSTU/LR8/app/controllers$
```