



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 № 8**

**Название:** Ruby on Rails

**Дисциплина:** Языки интернет-программирования

Студент

ИУ6-32Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

Джафаров Э.Э.

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Петрова Я.С.

(И.О. Фамилия)

Москва, 2022

### **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:

Число из  $n$  цифр является числом Армстронга, если сумма цифр этого числа, возведенных в степень  $n$  равна самому этому числу. Например:  $153 = 1^3 + 5^3 + 3^3$ . Написать программу, определяющую все числа Армстронга, состоящие из  $n$  цифр ( $n$  вводится с клавиатуры). Вывести на печать сами числа их количество.

## Решение:

### 1. Контроллер calc\_controller.rb

```
1 class CalcController < ApplicationController
2   def input
3   end
4
5   def show
6     @right_numbers = []
7     @a = params[:power].to_i
8     (10**(@a - 1)...(10**@a)).step(1) do |number|
9       @right_numbers.push(number) if find(number) == number
10    end
11  end
12
13  def find(number)
14    summa = 0
15    @a.times do
16      tri = number % 10
17      number /= 10
18      summa += tri**@a
19    end
20    summa
21  end
22 end
```

### 2. input.html.erb

```
1 <h1>Здравейте</h1>
2 <%= form_tag("/calc/show", :method => :get) do %>
3   <%= label_tag("Введите количество цифр в числе:") %>
4   </br>
5   <%= number_field_tag(:power, nil, in: 1..6, step:1)%>
6   </br>
7   <%= submit_tag("Yosh") %>
8   <% end %>
9
```

### 3. show.html.erb

```
<h1 align="center">Посчитали</h1>
<p align="center">Числа Армстронга:</p>
<table border=1 align="center">
  <tr>
    <td>Порядковый номер</td>
    <td> Число  </td>
  </tr>
  <% @right_numbers.each_with_index do |number, index| %>
    <tr>
      <td align="center"> <%= index + 1 %> </td>
      <td align="center"> <%= number %> </td>
    </tr>
  <% end %>
</table>
<p align="center">Количество чисел Армстронга: <%= @right_numbers.length %></p>
<a href="/>Вернуться
```

### 4. routes.rb

```
1  Rails.application.routes.draw do
2    root 'calc#input'
3    get 'calc/show'
4    # Define your application routes per the DSL in
5
6    # Defines the root path route ("/")
7    # root "articles#index"
8  end
9
```

## 5. Тест calc\_spec.rb

```
require 'rails_helper'

RSpec.describe "Calcs", type: :request do
  describe "GET /" do
    it "returns http success" do
      get "/"
      expect(response).to have_http_status(:success)
    end
  end

  describe "GET /show" do
    it "returns http success" do
      get "/calc/show"
      expect(response).to have_http_status(:success)
    end

    it 'return right values' do
      get "/calc/show", params: { :power => "3"}
      expect(assigns(:right_numbers)).to eq([153, 370, 371, 407])
    end
  end
end
```

## 6. Результат теста:

```
work/IPL-Laboratory8/spec$ rspec requests/calc_spec.rb
...
Finished in 0.5557 seconds (files took 7.78 seconds to load)
3 examples, 0 failures
```

## 7. Результат работы:

### 7.1 Страница ввода:

# Здрасьте

Введите количество цифр в числе:

## 7.2 Страница вывода:

# Посчитали

Числа Армстронга:

Порядковый номер	Число
1	153
2	370
3	371
4	407

Количество чисел Армстронга: 4

## 8. Проверка Rubocop:

```
work/IPL-Laboratory8/spec$ rspec requests/calc_spec.rb
...
Finished in 0.5557 seconds (files took 7.78 seconds to load)
3 examples, 0 failures
```

## Вывод:

Я познакомился с Ruby on rails и научился создавать простые контроллеры и обрабатывать данные от пользователя