=begin

ПРОГРАММА АНАЛИЗА ВВЕДЕННОГО ТЕКСТА.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

I раздел. Ruby.

Задачи:

- обработать ошибки ввода:

- пустой ввод;

- не английский язык;

- вывести исходный текст;

- посчитать количество знаков;

- посчитать количество пробелов;

- посчитать количество цифр;

- вывести введенные цифры строкой;

- вывести только четные цифры;

- вывести только нечетные цифры;

- сравнить цифры и вывести наибольшую;

- для наибольшей цифры посчитать факториал;

- посчитать частоту цифр;

- посчитать сумму использованных цифр;

- посчитать произведение использованных цифр;

- посчитать количество символов (не букв и цифр);

- вывести символы (не буквы и цифры);

- посчитать большие буквы;

- вывести большие буквы;

- посчитать маленькие буквы;

- вывести маленькие буквы;

- убрать из исходного текста цифры и символы (не буквы и цифры), сделать все буквы маленькими и вывести все буквы;

- инвертировать порядок букв и вывести;

- посчитать частоту букв и вывести на экран;

- вывести на экран частоту только упоминаемых букв;

- создать файл и сохранять (дополнять) в него результаты по мере выполнения заданий;

- создать файл с чеклистом и по мере выполнения заданий (прохождения тестов) отмечать в нем выполнение;

- все задания должны иметь тест;

- использовать github для хранения кода на всех этапах выполнения заданий;

- задания должны быть branch и при выполнении сливаться с master;

- код должен быть документирован.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

II раздел. Ruby on Rails.

Задачи:

Создать для проекта веб-интерфейс.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

III раздел. Условия выполнения.

Требования по использованию ruby:

- модули;

- классы и суперклассы;

- структуры и OStruct;

- блоки кода;

- наследование;

- примеси;

- методы:

- доступа (геттеры, сеттеры, асессоры);

- to\_s;

- super;

- initialize;

- Math;

- Time;

- Proc;

- lambda;

- диапазоны;

- массивы;

- хэши и символы;

- вложенный хэш;

- вложенный масссив;

- иттератор;

- условные операторы:

- if;

- unless;

- case;

- циклы:

- for;

- while;

- until;

- рекурсия;

- переменные:

-локальные;

- глобальные;

- класса;

- экземпляра;

- модификаторы доступа:

- public;

- private;

- protected.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IV раздел. План реализации.

Реализация:

1.

2. Создание хэша с ключами буквами и пустыми параметрами.

3.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

V раздел. Код.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

=end

hash\_var = Hash.new()

hash\_var = {first:1, second:2}

hash\_add = {third:3}

hash\_var.merge! = {first:1, second:2}

hash\_var.each{|key, volume| puts "#{key}: #{volume}"}

=begin

commite

=end

#commite

#branch

puts "Для подсчёта частоты букв \nв произвольной фразе \nвведите фразу на английском: "

text = gets

puts text

#предобработка текста

text.downcase!

#

freqs = {}

#("a".."z").each {|letter| freqs << letter:default}

#puts freqs

#freqs.default = 0

#подсчёт повторений букв

text.each\_char {|char| freqs [char] +=1 }

puts {text.lengh}

#вывод алфавита с частотой букв во фразе

("a".."z").each {|letter| puts "#{letter}: #{freqs[letter]}"}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ht = Hash.new

ht = {first:1, seconb:2}

for i in "a".."z"

time\_hash = {i, 0}

puts time\_hash

ht.merge! (time\_hash)

end

ht.each {|key, volume| puts "#{key}: #{volume}"}

hash\_var = Hash.new()

hash\_var = {first:1, second:2}

hash\_add = {third:3}

hash\_var.merge! = {first:1, second:2}

hash\_var.each{|key, volume| puts "#{key}: #{volume}"}

=begin

commite

=end

#commite