

Šiame faile yra užduotys nuo pirmos iki ketvirtos. Šias užduotis atlikau su rubymine programa per virtualbox.

```
main0.rb - Notepad
File Edit Format View Help
|=begin
Create a class Calculator, which performs addition and subtraction of two numbers at a time
adding 2 and 3 gives 5
adding 4 and -4 gives 0
Subtracting 2 from 5 gives 3
=end

class Calculator
  def add(a, b)
    a + b
  end

  def subtract(a, b)
    a - b
  end
end

object = Calculator.new
puts object.add(2,3)
puts object.add(4,-4)
puts object.subtract(5,2)
```

Šiame paveikslėlyje yra realizuotas skaičiuotuvas. Jis atlieka sudetis ir atimties skaičiavimus, sukuriame klasę ir paduodami skaičiai su puts object.

```
File Edit Format View Help
|=begin
Given an Array, return the elements that are present exactly once in the array.
returns [1,4,5], given [1,2,2,3,3,4,5]
returns [1,3], given [1,2,2,3,4,4]
=end

def non_duplicated_values(values)
  values.find_all { |x| values.count(x) == 1 }
end

array1 = [1,2,2,3,3,4,5]
array2 = [1,2,2,3,4,4]

puts "#{non_duplicated_values(array1)}"
puts "#{non_duplicated_values(array2)}"
```

Šiame paveikslėlyje yra kodo fragmentas, kurio pagalba yra atvaizduojami skaičiai kurie nesikartoja, tai realizuota non_duplicated_values pagalba.

```

main2.rb - Notepad
File Edit Format View Help
=begin
Given a sentence, return true if the sentence is a palindrome.
returns true for 'Race fast safe car'
returns true for 'Live not on Evil'
returns false for 'Yreka Bakey'
=end
def palindrome?(sentence)
  newsentence = sentence.downcase.gsub(" ", "")
  if newsentence == newsentence.reverse
    true
  else
    false
  end
end

puts palindrome?("Race fast safe car")
puts palindrome?("Live not on Evil")
puts palindrome?("Yreka Bakey")

```

Šiame paveikslėlyje yra ieškomas polindromas, tai yra žodis sakinyš ar kažkas kas iš prieko skaitosi taip pat kaip ir iš galo. Sakinyš yra sulyginamas su atvirkščiai parašytu sakiniu.

```

*main3.rb - Notepad
File Edit Format View Help
=begin
int main(int argc, const char * argv[]) {
  //rūšiavimas
  int array[11] = {10,12,23,43,12,33,2,1,11,33,6}; //11
  for (int i = 0; i < 11; i++) {
    for (int a = 0; a < 11; a++) {
      if(a + 1 < 11){
        if (array[a] < array[a+1]) {

          int temp = array[a];
          cout<<array[a]<<" "<<array[a+1]<<endl;
          array[a] = array[a+1];
          array[a+1] = temp;
        }
      }
    }
    cout<<"Ats"<<endl;
    for (int i = 0; i < 11; i++) {
      cout<<array[i]<<" ";
    }

    cout<<endl;
  }
}
=end
def sort(array)
  array.sort
end
array = [10,12,23,43,12,33,2,1,11,33,6]
puts "#{array}"
puts "#{sort(array)}"

```

Čia yra pateiktas skaičių rūšiavimo kodas, jo deka skaičiai kurie yra sekoje yra surišuojami nuo mažiausio iki didžiausio.


*main4.rb - Notepad

File Edit Format View Help

```
//pirminiai skaičiai
  int i = 0;
  int skaicius = 1;
  while (i < 20) {
    skaicius++;
    bool arPirminis = true;
    for (int j = 2; j < skaicius; j++) {
      if (skaicius % j == 0) {
        arPirminis = false;
        break;}}
    if (arPirminis) {
      i++;
      cout<<skaicius<<endl; } }
=end
def is_prime(num)
  n = 2
  while n < num
    return false if num % n == 0
    n += 1
  end
  true
end
array = Array.new
skaicius = 2
while skaicius <= 20
  if is_prime(skaicius) == true
    array << skaicius
  end
  skaicius = skaicius + 1
end
puts "#{array}"
```

Šiame kode yra ieškomi pirminiai skaičiai iki 20. Pirminis skaičius turi būti didesnis nei vienetą ir turi dalintis tik iš savęs ir iš vieneto. Paleidus šį kodą yra gaunama 2;3;5;7;11;13;17;19.

Trečia ir ketvirta

 mainnn0.rb - Notepad

File Edit Format View Help

```
# frozen_string_literal: true
# frozen_string_literal: true

class Pirminiai
  def prime?(num)
    n = 2
    while n < num
      return false if (num % n).zero?

      n += 1
    end
    true
  end

  def tikrinimas(number)
    array = []
    skaicius = 2
    while skaicius <= number
      array << skaicius if prime?(skaicius) == true
      skaicius += 1
    end
    array
  end
end

# object = Pirminiai.new
# puts object.tikrinimas(20).to_s
```

Čia pavaizduotas antros užduoties kodas kuris perrašytas *Object-Oriented principu*. Atliekamas tas pats pirminio skaičiaus ieškojimas tik objekto pagalba

```

mainnn1.rb - Notepad
File Edit Format View Help
# frozen_string_literal: true

class Rykiavimas
  def sort(array)
    array.sort
  end
end

array = [10, 12, 23, 43, 12, 33, 2, 1, 11, 33, 6]
object = Rykiavimas.new
puts array.to_s
puts object.sort(array).to_s

```

Čia pavaizduotas antros užduoties kodas kuris perrašytas *Object-Oriented principu*. Sukuriamas objektas kurio pagalba yra išrikiuojami skaičiai.

```

require_relative 'mainnn0'
require 'test/unit'

class TestPirminiai < Test::Unit::TestCase
  def test_prime?
    assert_equal(true, Pirminiai.new.prime?(5))
    assert_equal(true, Pirminiai.new.prime?(11))
    assert_equal(false, Pirminiai.new.prime?(8))
  end

  def test_tikrinimas
    assert_equal([2, 3, 5], Pirminiai.new.tikrinimas(5))
    assert_not_equal([2, 4, 5], Pirminiai.new.tikrinimas(5))
  end
end

```

Šiame kode yra parašytas testas, kuris turi parodyti ar skaičiai yra pirminiai, tai `assert_equal(true, Pirminiai.new.prime?(5))` turime gauti true, `assert_equal(true, Pirminiai.new.prime?(11))` turime gauti true, `assert_equal(false, Pirminiai.new.prime?(8))` turime gauti false nes skaičius nėra pirminis