

## TASK 1

Prieš rubocop:

```
1  # The Exam Class
2  class Exam
3    print 'Iveskite, studento gauta ivertinima: '
4    a = Integer(gets) # a yra egzamino gautas ivertinimas
5
6    if a >= 5 && a < 10 # salyga tikrina ar iverstas ivertinimas nera mazesnis nei 5 ir nera
    didesnis nei 10
7      print 'Egzaminas yra islaikytas.' # jei pirmoji if salyga atitiko reikalavimus gauname
      toki pranesima
8    elsif a >= 0 && a < 5 # salyga tikrina ar skaicius nera mazesnis uz 0 bei nera didesnis
    uz 5
9      print 'Egzaminas yra neislaikytas.' # jei antroji if salyga atitiko reikalavimus
      gauname toki pranesima
10   elsif a.negative? || a > 10 # salyga tikrina ar iverstas ivertinimas nera mazesnis uz 0
      ir nera didesnis uz 10
11     print 'Skaicius negali buti mazesnis nei 0 arba didesnis nei 10.'
12     # jei trecioji if salyga atitiko reikalavimus gauname toki pranesima
13   end
14 end
15
```

Rubocop:

```
dziugas@MSI:~$ rubocop -A /mnt/c/Users/Džiugas/Desktop/First.rb
Inspecting 1 file
C

Offenses:

/mnt/c/Users/Džiugas/Desktop/First.rb:1:1: C: Naming/FileName: The name of this source file (First.rb) should use snake_
case.
# frozen_string_literal: true
^
1 file inspected, 1 offense detected
```

Po rubocop:

```

1  # frozen_string_literal: true
2
3  # The Exam Class
4  class Exam
5    print 'Iveskite, studento gauta ivertinima: '
6    a = Integer(gets) # a yra egzamino gautas ivertinimas
7
8    if a >= 5 && a < 10 # salyga tikrina ar iverstas ivertinimas nera mazesnis nei 5 ir nera
    didesnis nei 10
9      print 'Egzaminas yra islaikytas.' # jei pirmoji if salyga atitiko reikalavimus gauname
      toki pranesima
10   elsif a >= 0 && a < 5 # salyga tikrina ar skaicius nera mazesnis uz 0 bei nera didesnis
    uz 5
11     print 'Egzaminas yra neislaikytas.' # jei antroji if salyga atitiko reikalavimus
      gauname toki pranesima
12   elsif a.negative? || a > 10 # salyga tikrina ar iverstas ivertinimas nera mazesnis uz 0
      ir nera didesnis uz 10
13     print 'Skaicius negali buti mazesnis nei 0 arba didesnis nei 10.'
14     # jei trecioji if salyga atitiko reikalavimus gauname toki pranesima
15   end
16 end
17

```

## TASK 2

Prieš rubocop:

```

1  #The Math Class
2  class Math
3    print 'Iveskite, intervalo pradzia: '
4    a = Integer(gets) # a yra iversta intervalo pradzia
5    print 'Iveskite, intervalo pabaiga: '
6    b = Integer(gets) # b yra iversta intervalo pabaiga
7    integer_suma = 0 # naujas kintamasis skirtas
      skaiciuoti lyginiu skaiciu suma
8    (a..b).each do |i| # ciklas ima visas reiksmes nuo a iki b didindamas vienetu
9      if i.even?
10        # if salyga tikrina ar intervalo skaiciu dalindami is dvieju gausime liekana 0, jei
          liekana=0 skaicius sveikas
11        integer_suma += i # if salygai pasitvirtinus pradedame sumuoti lyginius skaicius
12      end
13    end
14    puts("Intervalo pradzia ir pabaiga yra: #{a};#{b}") # isvedame i ekrana intervalo
      pradzia ir pabaiga
15    puts("Lyginiu skaiciu suma yra: #{integer_suma}") # isvedame i ekrana intervalo
      lyginiu skaiciu suma
16  end

```

Rubocop:

```
džiugas@MSI:~$ rubocop -A /mnt/c/Users/Džiugas/Desktop/Second.rb
Inspecting 1 file
C

Offenses:

/mnt/c/Users/Džiugas/Desktop/Second.rb:1:1: C: Naming/FileName: The name of this source file (Second.rb) should use snake_case.
# frozen_string_literal: true
^
/mnt/c/Users/Džiugas/Desktop/Second.rb:1:1: C: [Corrected] Style/FrozenStringLiteralComment: Missing frozen string literal comment.
# The Math Class
^
/mnt/c/Users/Džiugas/Desktop/Second.rb:2:1: C: [Corrected] Layout/EmptyLineAfterMagicComment: Add an empty line after magic comments.
# The Math Class
^

1 file inspected, 3 offenses detected, 2 offenses corrected
```

Po rubocop:

```
1 # frozen_string_literal: true
2
3 # The Math Class
4 class Math
5   print 'Iveskite, intervalo pradzia: '
6   a = Integer(gets) # a yra iversta intervalo pradzia
7   print 'Iveskite, intervalo pabaiga: '
8   b = Integer(gets) # b yra iversta intervalo pabaiga
9   integer_suma = 0 # naujas kintamasis skirtas
10  # skaiciuoti lyginiu skaiciu suma
11  (a..b).each do |i| # ciklas ima visas reiksmes nuo a iki b didindamas vienetu
12    if i.even?
13      # if salyga tikrina ar intervalo skaiciu dalindami is dvieju gausime liekana 0, jei
14      # liekana=0 skaicius sveikas
15      integer_suma += i # if salygai pasitvirtinus pradedame sumuoti lyginius skaicius
16    end
17  end
18  puts("Intervalo pradzia ir pabaiga yra: [{a};{b}]") # isvedame i ekrana intervalo
19  # pradzia ir pabaiga
20  puts("Lyginiu skaiciu suma yra: {integer_suma}") # isvedame i ekrana intervalo
21  # lyginiu skaiciu suma
22 end
```