1 uzduotis – Sum of Negatives

Uzduotis-

Create a function that takes an array of positive and negative numbers. Return an array where the first element is the **count** of positive numbers and the second element is the **sum** of negative numbers.

Examples

```
CountPosSumNeg([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -11, -12, -13, -14, -15]) ➞ [
// There are a total of 10 positive numbers.
// The sum of all negative numbers equals -65.

CountPosSumNeg([92, 6, 73, -77, 81, -90, 99, 8, -85, 34]) ➞ [7, -252]

CountPosSumNeg([91, -4, 80, -73, -28]) ➞ [2, -105]

CountPosSumNeg([]) ➞ []
```

Notes

- If given an empty array, return an empty array: `[]`
- Cast sum to int, don't mind the remaining decimal places.
- 0 is not positive.

SUGGEST EDIT

C# source kodas –

```
public class Program
{
    public static int[] CountPosSumNeg(double[] arr)
    {
      if(arr.Length == 0 || arr == null)
      {
        return new int[2];
      }
      int[] result = new int[] {0, 0};
      for(int i = 0; i < arr.Length; i++)
      {
        if(arr[i] > 0)
        {
          result[0]++;
        }
        else
        {
          result[1] += (int)arr[i];
        }
      }

      return result;
    }
}
```

Ruby kodas :

```ruby
#Sukuriama nauja Sum klase, kurios viduje deklaruotas self metodas
class Sum
  #Sukuriamas naujas klases metodas, kuris apskaiciuoja neigiamu skaiciu suma,
  Skaicius gauna array pavidalu.
  def self.SumOfNegatives(array)
    #Per kiekviena array elementa sukamas for ciklas ir tikrinama ar skaicius i yra
    teigiamas arba neigiamas
    #Jei skaicius neigiamas yra pridedamas prie atsakymo sumos
   for i in array
    if i<0
      sum=sum.to_i+i.to_i
    end
    end
    #Atspausdina atsakyma ekrane
  puts sum
  end
end
array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -11, -12, -13, -14, -15]
#Kreipinys i sum klases self SumOfNegatives metoda paduodant array
Sum.SumOfNegatives(array)
```

Console:

```
> ruby main.rb
-65
>
```

2 uzduotis – Array of multiples

```ruby
#Sukuriama nauja klase
class ArrayOfMultiples
  #Aprasomas naujas self multiply metodas priimantis du skaicius
  def self.multiply(firstNr, secondNr)
    #atsakymo masyvas, kuris laiko skaiciu sandaugas
    answer = []
    sum = 0
    i = 0
    #While ciklas yra vykdomas kol i yra mazesnis uz pirma skaiciu, kiekvienos iteracijos metu i yra padidinamas
    while i < firstNr.to_i do
     #Prie sum pridedamas antras skaicius kiekviena iteracija.
  sum+=secondNr.to_i
      #kiekvienas sum pushinamas i array tokiu budu, gaunama kiekvieno skaiciaus sandauga
  answer.push(sum)
  i += 1
  end
    #atspausdinamas sandaugu masyvas
  puts "Array of multiples:"
  puts answer
  end
end


#Nuskaitomi skaiciai is consolines eilutes
require "readline"
puts "Input first number:"
firstNr = gets
puts "Input second number:"
secondNr = gets
#Kreipinys i klases multiply metoda perduodant pirma ir antra skaiciu
#Pirmas skaicius - Kiek kartu bus antras skaicius padaugintas
#Antras skaicius - Skaicius, kuri daugins.
ArrayOfMultiples.multiply(firstNr, secondNr)
```

Create a function that takes two numbers as arguments (`num`, `length`) and returns an array of multiples of `num` up to `length`.

Examples

```
ArrayOfMultiples(7, 5) → [7, 14, 21, 28, 35]

ArrayOfMultiples(12, 10) → [12, 24, 36, 48, 60, 72, 84, 96, 108, 120]

ArrayOfMultiples(17, 6) → [17, 34, 51, 68, 85, 102]
```

Notes

Notice that `num` is also included in the returned array.

SUGGEST EDIT

Source kodas c#

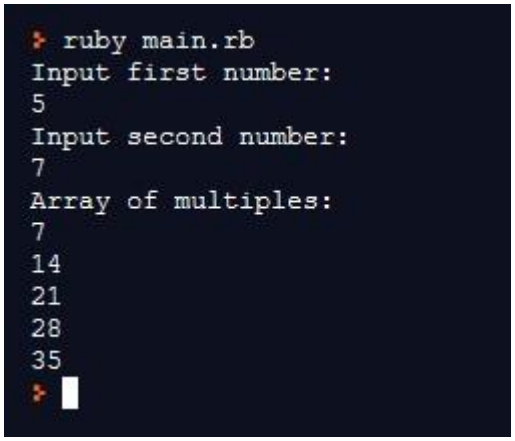
```
using System;

public class Program {
    public static int[] ArrayOfMultiples(int num, int length)
    {
        int[] multiples = new int[length];

        for(int i = 0; i < length; i++) {
            multiples[i] = num * (i + 1);
        }
        return multiples;
    }
}
```

Ruby kodas

Console :



3 uzduotis Uzduotis :

Create a function that returns `true` if a number is a palindrome.

Examples

```
IsPalindrome(838) → true

IsPalindrome(4433) → false

IsPalindrome(443344) → true
```

Notes

- A palindrome is a number that remains the same when reversed.
- Bonus: Try solving this without turning the number into a string.

C# source kodas:

```
public class Program
{
    public static bool IsPalindrome(int num)
    {
      string str = num.ToString();
      for (int i = 0; i < str.Length / 2; i++)
    {
        if (str[i] != str[str.Length - i - 1])
            return false;
    }
    return true;
    }
}
```

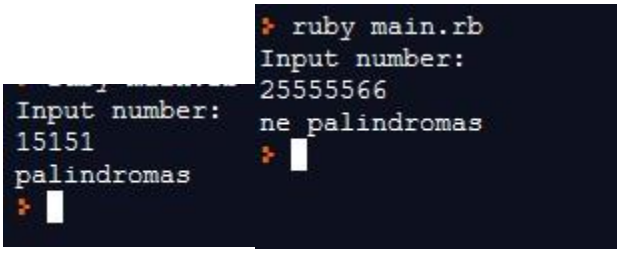
Ruby codas:

```ruby
#Aprasoma Number klase
class Number
  #Metodas kuris apvercia skaiciu
  def self.Polyndrome(firstNr)
  reversed =0
    #issaugomas pirmas int skaicius, kad veliau butu su kuo palyginti
temp = firstNr.to_i
#While ciklas sukasi kol skaicius yra didesnis uz 0
    #su kiekviena iteracija skaicius yra prie apversto skaiciaus reversed yra pridedama paskutine firstNr rei
while firstNr.to_i > 0 do
  #padauginama is 10, kad butu galima prideti paskutine firstNr reiksme
    reversed = reversed *10
  #Pridedama paskutine firstNr skaiciaus reiksme
    reversed = reversed + (firstNr.to_i%10)
  #nuiimama panaudota paskutine reiksme
    firstNr = firstNr.to_i/10
end
    #Kreipinys i PrintInfo klase, kurioje atspausdinamas rezultatas. Paduodant temp ir reversed
    #temp - pradinis issaugotas skaicius
    #reversed - apverstas pradinis skaicius
PrintInfo.PutAnswer(reversed, temp)
end


class PrintInfo
  #metodas kuris tikrina ar skaicius yra polindromas
  def self.PutAnswer(reversed,temp)
    #tikrinama ar skaicius reversed yra polindromas palyginant su temp
  if reversed==temp
    puts "palindromas"
 else
  puts "ne palindromas"
      end
  end
end
end

#Nuskaito skaiciu is consolines eilutes
require "readline"
firstNr=0
puts "Input number:"
firstNr = gets
#perduoda skaiciu i Number klases Polundrome metoda
Number.Polyndrome(firstNr)
```

Console:



4uzduotis –

Uzduotis -

Create a Fact method that receives a **non-negative** integer and `returns` the factorial of that number.

Examples

```
Fact(1) → 1

Fact(3) → 6

Fact(6) → 720
```

Notes

- Consider that 0! = 1.
- You should return a `long` data type number.

C# source kodas

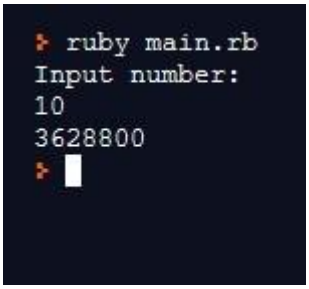
```
public class Program
{
  public static long Fact(int n)
    => n <= 1 ? 1 : n * Fact(n - 1);
}
```

Ruby codas

```ruby
#Klase kuri apskaiciuoja skaicisu faktoriala
class Integer
  def fact
    #nuo 1 iki pasirinkto skaiciaus
    #Kiekviena iteracija skaicius yra padauginamas
    (1..self).reduce(:*) || 1
  end
end
#Nuskaitomas skaicius is consolines eilutes
require "readline"
firstNr=0
puts "Input number:"
firstNr = gets

#Parasomas atsakymas skaicium kreipiantis i klases Integer Fact metoda.
puts firstNr.to_i.fact
```

Console:



RUBOCOP:
Pirma kart paleidus rubocop:

Offenses:

2pd/fourth.rb:1:1: C: Style/Documentation: Missing top-level class documentation comment.
class Integer
^^^^^
2pd/fourth.rb:1:1: C: Style/FrozenStringLiteralComment: Missing frozen string literal comment.
class Integer
^
2pd/main.rb:1:1: C: Style/Documentation: Missing top-level class documentation comment.
class Sum
^^^^^

```
2pd/main.rb:1:1: C: Style/FrozenStringLiteralComment: Missing frozen string literal comment.
class Sum
^
2pd/main.rb:2:1: C: Layout/IndentationWidth: Use 2 (not 0) spaces for indentation.
def self.SumOfNegatives(array)

2pd/main.rb:2:10: C: Naming/MethodName: Use snake_case for method names.
def self.SumOfNegatives(array)
       ^^^^^^^^^^^^^^
2pd/main.rb:3:1: C: Layout/IndentationWidth: Use 2 (not 0) spaces for indentation.
for i in array

2pd/main.rb:3:1: C: Style/For: Prefer each over for.
for i in array ...
^^^^^^^^^^^^^^
2pd/main.rb:4:1: C: Layout/IndentationWidth: Use 2 (not 0) spaces for indentation.
if i<0

2pd/main.rb:4:1: C: Style/IfUnlessModifier: Favor modifier if usage when having a single-line body. Another good alternative is the usage of control flow &&/||.
if i<0
^^
2pd/main.rb:4:4: C: Style/NumericPredicate: Use i.negative? instead of i<0.
if i<0
  ^^^
2pd/main.rb:4:5: C: Layout/SpaceAroundOperators: Surrounding space missing for operator <.
if i<0
   ^
2pd/main.rb:5:1: C: Layout/IndentationWidth: Use 2 (not 0) spaces for indentation.
sum=sum.to_i+i.to_i

2pd/main.rb:5:4: C: Layout/SpaceAroundOperators: Surrounding space missing for operator =.
sum=sum.to_i+i.to_i
  ^
2pd/main.rb:5:13: C: Layout/SpaceAroundOperators: Surrounding space missing for operator +.
sum=sum.to_i+i.to_i
         ^
2pd/main.rb:13:1: C: Layout/TrailingEmptyLines: 1 trailing blank lines detected.
2pd/second.rb:1:1: C: Style/Documentation: Missing top-level class documentation comment.
class ArrayOfMultiples
^^^^^
2pd/second.rb:1:1: C: Style/FrozenStringLiteralComment: Missing frozen string literal comment.
class ArrayOfMultiples
^
```

```
2pd/second.rb:6:29: C: Style/WhileUntilDo: Do not use do with multi-line while.
  while i < first_nr.to_i do
                      ^^
2pd/second.rb:7:3: C: Layout/IndentationWidth: Use 2 (not -2) spaces for indentation.
 sum+=second_nr.to_i
 ^^
2pd/second.rb:7:6: C: Layout/SpaceAroundOperators: Surrounding space missing for operator +=.
 sum+=second_nr.to_i
   ^^
2pd/second.rb:10:3: W: Layout/EndAlignment: end at 10, 2 is not aligned with while at 6, 4.
 end
 ^^^
2pd/second.rb:11:3: C: Layout/IndentationConsistency: Inconsistent indentation detected.
 puts 'Array of multiples:'
 ^^^^^^^^^^^^^^^^^^^^^^^^^^
2pd/second.rb:12:3: C: Layout/IndentationConsistency: Inconsistent indentation detected.
 puts answer
 ^^^^^^^^^^^
2pd/third.rb:1:1: C: Style/Documentation: Missing top-level class documentation comment.
class Number
^^^^^
2pd/third.rb:1:1: C: Style/FrozenStringLiteralComment: Missing frozen string literal comment.
class Number
^
2pd/third.rb:2:12: C: Naming/MethodName: Use snake_case for method names.
 def self.Polyndrome(first_nr)
          ^^^^^^^^^^
2pd/third.rb:3:3: C: Layout/IndentationWidth: Use 2 (not 0) spaces for indentation.
 reversed =0

2pd/third.rb:3:12: C: Layout/SpaceAroundOperators: Surrounding space missing for operator =.
 reversed =0
          ^
2pd/third.rb:4:1: C: Layout/IndentationConsistency: Inconsistent indentation detected.
temp = first_nr.to_i
^^^^^^^^^^^^^^^^^^^
2pd/third.rb:5:1: C: Layout/IndentationConsistency: Inconsistent indentation detected.
while first_nr.to_i > 0 do ...
^^^^^^^^^^^^^^^^^^^^^^^^^^^
2pd/third.rb:5:7: C: Style/NumericPredicate: Use first_nr.to_i.positive? instead of first_nr.to_i > 0.
while first_nr.to_i > 0 do
    ^^^^^^^^^^^^^^^^^
2pd/third.rb:5:25: C: Style/WhileUntilDo: Do not use do with multi-line while.
while first_nr.to_i > 0 do
```

```
                ^^
2pd/third.rb:6:1: C: Layout/IndentationWidth: Use 2 (not 4) spaces for indentation.
  reversed = reversed * 10
^^^^
2pd/third.rb:6:5: C: Style/SelfAssignment: Use self-assignment shorthand *=.
  reversed = reversed * 10
  ^^^^^^^^^^^^^^^^^^^^^^^^
2pd/third.rb:7:5: C: Style/SelfAssignment: Use self-assignment shorthand +=.
  reversed = reversed + (first_nr.to_i % 10)
  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
2pd/third.rb:8:34: C: Layout/TrailingWhitespace: Trailing whitespace detected.
  first_nr = first_nr.to_i / 10
                          ^
2pd/third.rb:11:1: W: Layout/DefEndAlignment: end at 11, 0 is not aligned with def at 2, 2.
end
^^^
2pd/third.rb:12:1: C: Layout/IndentationConsistency: Inconsistent indentation detected.
class PrintInfo ...
^^^^^^^^^^^^^^^
2pd/third.rb:12:1: C: Layout/IndentationWidth: Use 2 (not 0) spaces for indentation.
class PrintInfo

2pd/third.rb:12:1: C: Style/Documentation: Missing top-level class documentation comment.
class PrintInfo
^^^^^
2pd/third.rb:14:3: C: Layout/IndentationWidth: Use 2 (not 0) spaces for indentation.
 if reversed==temp

2pd/third.rb:14:14: C: Layout/SpaceAroundOperators: Surrounding space missing for operator ==.
 if reversed==temp
         ^^
2pd/third.rb:16:7: C: Layout/TrailingWhitespace: Trailing whitespace detected.
 else
    ^
2pd/third.rb:18:7: W: Layout/EndAlignment: end at 18, 6 is not aligned with if at 14, 2.
    end
    ^^^

  4        files inspected, 45 offenses detected, 42 offenses auto-correctable
```

po sutvarkymo:

```
2pd/fourth.rb:3:1: C: Style/Documentation: Missing top-level class documentation comment.
class Integer
```

^^^^^
2pd/main.rb:3:1: C: Style/Documentation: Missing top-level class documentation comment.
class Sum
^^^^^
2pd/main.rb:6:7: W: Lint/UselessAssignment: Useless assignment to variable - sum.
    sum += i.to_i if i.negative?
    ^^^
2pd/second.rb:3:1: C: Style/Documentation: Missing top-level class documentation comment.
class ArrayOfMultiples
^^^^^
2pd/third.rb:3:1: C: Style/Documentation: Missing top-level class documentation comment.
class Number
^^^^^
2pd/third.rb:14:3: C: Style/Documentation: Missing top-level class documentation comment.
  class PrintInfo

Uzduociu kodai po sutvarkyto kodo su Robocop::

Main.rb

```ruby
# frozen_string_literal: true

class Sum
  def self.sum_of_negatives(array)
    array.each do |i|
      sum = sum.to_i + i.to_i if i.negative?
    end
    puts sum
  end
end
array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -11, -12, -13, -14, -15]
Sum.sum_of_negatives(array)

```

Second.rb

```ruby
# frozen_string_literal: true

class ArrayOfMultiples
  def self.multiply(first_nr, second_nr)
    answer = []
    sum = 0
    i = 0
    while i < first_nr.to_i
      sum += second_nr.to_i
      answer.push(sum)
      i += 1
    end
    puts 'Array of multiples:'
    puts answer
  end
end
require 'readline'
puts 'Input first number:'
first_nr = gets
puts 'Input second number:'
second_nr = gets
ArrayOfMultiples.multiply(first_nr, second_nr)

```

Third.rb

```ruby
# frozen_string_literal: true

class Number
  def self.polyndrome(first_nr)
    reversed = 0
    temp = first_nr.to_i
    while first_nr.to_i.positive?
      reversed *= 10
      reversed += (first_nr.to_i % 10)
      first_nr = first_nr.to_i / 10
    end
    PrintInfo.put_answer(reversed, temp)
  end
  class PrintInfo
    def self.put_answer(reversed, temp)
      if reversed == temp
        puts 'palindromas'
      else
        puts 'ne palindromas'
      end
    end
  end
end
require 'readline'
puts 'Input number:'
first_nr = gets
Number.polyndrome(first_nr)
```

Fourth.rb

```ruby
# frozen_string_literal: true

class Integer
  def fact
    (1..self).reduce(:*) || 1
  end
end
require 'readline'
puts 'Input number:'
first_nr = gets
puts first_nr.to_i.fact
```