Matthew Grandi

9/11/2018

CSCI 602-81 Fall 2018

Homework 1 – Intro to Ruby

## **Ruby\_intro.rb**

|  |
| --- |
| # Part 1 |
|  |  |
|  | def sum(array) |
|  | result = 0 |
|  | array.each do |num| |
|  | result += num |
|  | end |
|  | return result |
|  | end |
|  |  |
|  | def max\_2\_sum(array) |
|  | result = 0 |
|  | #initialize temporary numbers as lower than you'd expect from input. |
|  | #Should probably use some sort of constant here. |
|  | num1 = -99999 |
|  | num2 = -99999 |
|  | if array.length > 2 |
|  | # for each element in the array, determine if it is greater than currently |
|  | # stored num1 and num2, or greater than num2 only. This keeps track of the |
|  | # two greatest numbers in the array. |
|  | array.each do |num| |
|  | if num > num1 && num > num2 |
|  | num1 = num |
|  | elsif num > num2 |
|  | num2 = num |
|  | else |
|  | next |
|  | end |
|  | end |
|  | elsif array.empty? |
|  | return 0 |
|  | elsif array.length == 1 |
|  | return array[0] |
|  | end |
|  | result = num1 + num2 |
|  | return result |
|  | end |
|  |  |
|  | def sum\_to\_n?(array, n) |
|  | # create a new array to hold the sums of elements in the input array |
|  | temp = Array.new |
|  | # Check conditions outlined in the assignment |
|  | if array.empty? || array.length == 1 |
|  | return false |
|  | else |
|  | # nested loop to add together each pair of elements in the array |
|  | for i in 0..array.length |
|  | for x in i+1..array.length-1 |
|  | # Store the sum of index i and i+1 in the temp array |
|  | temp.push(array[i] + array[x]) |
|  | end |
|  | end |
|  | end |
|  | # check if input n exists in the temp array, return true or false. |
|  | return temp.include? n |
|  | end |
|  |  |
|  |  |
|  | # Part 2 |
|  |  |
|  | def hello(name) |
|  | return "Hello, #{name}" |
|  | end |
|  |  |
|  | def starts\_with\_consonant?(s) |
|  | # Regex here checks if the first item is an alpha (letter). |
|  | if s[0] =~ /[[:alpha:]]/ |
|  | # Regex here checks if the first item is NOT a vowel. |
|  | if s[0] !~/[[aeiouAEIOU]]/ |
|  | return true |
|  | end |
|  | else return false |
|  | end |
|  | end |
|  |  |
|  | def binary\_multiple\_of\_4?(s) |
|  | # regex here = from start of string, any amount of zeroes or ones, followed by |
|  | # a group of two zeroes at the end of the string. Special exception for single |
|  | # zero. Zero is a multiple of every number, but I can't figure out regex to |
|  | # filter it out. |
|  | if s =~ /^[01]\*(00)$/ || s == "0" |
|  | return true |
|  | else |
|  | return false |
|  | end |
|  |  |
|  | end |
|  |  |
|  | # Part 3 |
|  |  |
|  | class BookInStock |
|  |  |
|  | attr\_accessor :isbn |
|  | attr\_accessor :price |
|  |  |
|  | def initialize(isbn, price) |
|  | @isbn = isbn |
|  | @price = price |
|  | raise ArgumentError if isbn.empty? || price <= 0 |
|  | end |
|  |  |
|  |  |
|  | def price\_as\_string |
|  | # Uses string formatter to convert price to two decimal points. Adds a $. |
|  | return "$" + '%.2f' % @price |
|  | end |
|  |  |
|  | end |

## ***Test Results***

ec2-user:~/environment/hw-ruby-intro-1 (master) $ rspec spec/part1\_spec.rb

**Ruby intro part 1**

**#sum**

should be defined

returns correct sum [20 points]

works on the empty array [10 points]

**#max\_2\_sum**

should be defined

returns the correct sum [7 points]

works even if 2 largest values are the same [3 points]

returns zero if array is empty [10 points]

returns value of the element if just one element [10 points]

**#sum\_to\_n**

should be defined

returns true when any two elements sum to the second argument [30 points]

returns false for any single element array [5 points]

returns false for an empty array [5 points]

Finished in 0.01204 seconds (files took 0.09283 seconds to load)

12 examples, 0 failures

ec2-user:~/environment/hw-ruby-intro-1 (master) $ rspec spec/part2\_spec.rb

**#hello**

should be defined

The hello method returns the correct string [30 points]

**#starts\_with\_consonant?**

should be defined

classifies true cases [10 points]

classifies false cases [10 points]

works on the empty string [5 points]

works on nonletters [5 points]

**#binary\_multiple\_of\_4?**

should be defined

classifies valid binary numbers [30 points]

rejects invalid binary numbers [10 points]

Finished in 0.01015 seconds (files took 0.09578 seconds to load)

10 examples, 0 failures

ec2-user:~/environment/hw-ruby-intro-1 (master) $ rspec spec/part3\_spec.rb

**BookInStock**

should be defined

getters and setters

should set ISBN [10 points]

should set price [10 points]

should be able to change ISBN [10 points]

should be able to change price [10 points]

**constructor**

should reject invalid ISBN number [10 points]

should reject zero price [10 points]

should reject negative price [10 points]

**#price\_as\_string**

should be defined

should display 33.95 as "$33.95" [10 points]

should display 1.1 as $1.10 [10 points]

should display 20 as $20.00 [10 points]

Finished in 0.01094 seconds (files took 0.09306 seconds to load)

12 examples, 0 failures

## ***Screenshot***

