**NAME: VISHWANTH P**

**REGNO: 21MIS1117**

CAT – 1 Solution

Question – 1

def generate\_temp\_data

  temps = {}

  (1..365).each do |day|

    temps[day] = rand(15..35)

  end

  temps

end

def calc\_avg\_temp(temps)

  total\_temp = temps.values.sum

  total\_temp.to\_f / temps.size

end

def find\_temp\_extremes(temps)

  hot\_day, hot\_temp = temps.max\_by { |\_day, temp| temp }

  cold\_day, cold\_temp = temps.min\_by { |\_day, temp| temp }

  { hot\_day: hot\_day, hot\_temp: hot\_temp, cold\_day: cold\_day, cold\_temp: cold\_temp }

end

def calc\_monthly\_avg(temps)

  days\_in\_month = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

  monthly\_avg = {}

  day = 1

  days\_in\_month.each\_with\_index do |days, idx|

    month\_temps = temps.slice(day, days).values

    monthly\_avg[idx + 1] = month\_temps.sum.to\_f / days

    day += days

  end

  monthly\_avg

end

def find\_long\_heatwave(temps)

  heatwave = 0

  max\_wave = 0

  temps.each\_value do |temp|

    if temp > 30

      heatwave += 1

    else

      max\_wave = [max\_wave, heatwave].max

      heatwave = 0

    end

  end

  max\_wave

end

def find\_long\_cold\_spell(temps)

  cold\_spell = 0

  max\_spell = 0

  temps.each\_value do |temp|

    if temp < 20

      cold\_spell += 1

    else

      max\_spell = [max\_spell, cold\_spell].max

      cold\_spell = 0

    end

  end

  max\_spell

end

def find\_hot\_month(monthly\_avg)

  monthly\_avg.max\_by { |\_month, avg\_temp| avg\_temp }.first

end

temps = generate\_temp\_data

avg\_temp = calc\_avg\_temp(temps)

extremes = find\_temp\_extremes(temps)

monthly\_avg = calc\_monthly\_avg(temps)

long\_heatwave = find\_long\_heatwave(temps)

long\_cold\_spell = find\_long\_cold\_spell(temps)

hot\_month = find\_hot\_month(monthly\_avg)

puts "Average Temp: #{avg\_temp}"

puts "Hottest Day: #{extremes[:hot\_day]} (#{extremes[:hot\_temp]}°C)"

puts "Coldest Day: #{extremes[:cold\_day]} (#{extremes[:cold\_temp]}°C)"

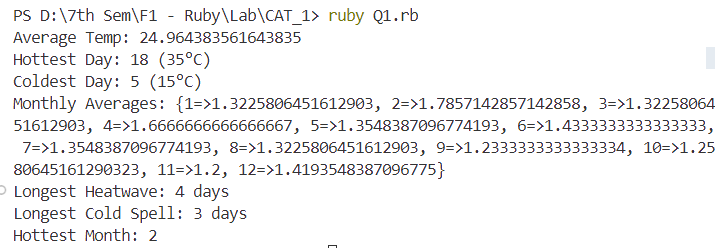
puts "Monthly Averages: #{monthly\_avg}"

puts "Longest Heatwave: #{long\_heatwave} days"

puts "Longest Cold Spell: #{long\_cold\_spell} days"

puts "Hottest Month: #{hot\_month}"

Output



Question – 2

class NumAnalyzer

  def find\_heads(nums, n)

    puts "Head numbers: "

    (1...n - 1).each do |i|

      if nums[i] > nums[i - 1] && nums[i] > nums[i + 1]

        puts nums[i]

      end

    end

  end

  def find\_max\_pair(nums, n)

    max\_pair = [nums[0], nums[1]]

    max\_sum = nums[0] + nums[1]

    (0...n - 1).each do |i|

      (i + 1...n).each do |j|

        cur\_sum = nums[i] + nums[j]

        if cur\_sum > max\_sum

          max\_sum = cur\_sum

          max\_pair = [nums[i], nums[j]]

        end

      end

    end

    max\_pair

  end

end

puts "Enter number of elements: "

n = gets.chomp.to\_i

nums = []

puts "Enter elements: "

n.times do

  nums << gets.chomp.to\_i

end

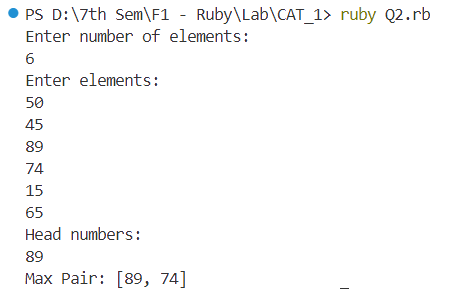
analyzer = NumAnalyzer.new

analyzer.find\_heads(nums, n)

max\_pair = analyzer.find\_max\_pair(nums, n)

puts "Max Pair: #{max\_pair}"

Output



Question – 3

class DynamicDispatcher

  def method\_missing(method\_name, \*args)

    if method\_name.to\_s.start\_with?("calculate")

      operation = method\_name.to\_s.split("calculate")[1]

      perform\_calculation(operation, args)

    else

      super

    end

  end

  def respond\_to\_missing?(method\_name, include\_private = false)

    method\_name.to\_s.start\_with?("calculate") || super

  end

  private

  def perform\_calculation(operation, args)

    case operation

    when "Factorial"

      puts factorial(args[0])

    when "Square"

      puts square(args[0])

    else

      puts "Unsupported operation: #{operation}"

    end

  end

  def factorial(n)

    return 1 if n == 0

    n \* factorial(n - 1)

  end

  def square(n)

    n \* n

  end

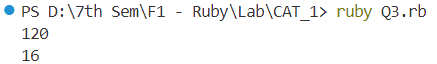
end

dispatcher = DynamicDispatcher.new

dispatcher.calculateFactorial(5)

dispatcher.calculateSquare(4)

Output



Question – 4

class BracketChecker

  def balanced\_parentheses(str)

    pairs = { '(' => ')', '{' => '}', '[' => ']', '<' => '>' }

    stack = []

    str.each\_char.with\_index do |char, idx|

      if pairs.key?(char)

        stack.push([char, idx])

      elsif pairs.value?(char)

        return "Unbalanced at position: #{idx + 1}" if stack.empty? || pairs[stack.last[0]] != char

        stack.pop

      end

    end

    stack.empty? ? true : "Unbalanced at position: #{stack.last[1] + 1}"

  end

  def evaluate\_expression(str)

    return balanced\_parentheses(str) unless balanced\_parentheses(str) == true

    exprs = str.scan(/\([^)]+\)/)

    exprs.each do |expr|

      result = eval(expr[1..-2])

      str.sub!(expr, result.to\_s)

    end

    str

  end

end

checker = BracketChecker.new

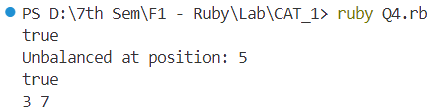
puts checker.balanced\_parentheses("(11)")

puts checker.balanced\_parentheses("|(|)])")

puts checker.balanced\_parentheses("<(1+2) (3+4)>")

puts checker.evaluate\_expression("(1+2) (3+4)")

Output



Question – 5

require 'fileutils'

class WordCounter

  def initialize(filename)

    @filename = filename

    @word\_counts = {}

  end

  def analyze\_word\_usage

    read\_file

    count\_words

    output\_results

  end

  private

  def read\_file

    File.open(@filename, 'r') do |file|

      @text = file.read

    end

  end

  def count\_words

    words = @text.downcase.gsub(/[^a-z\s]/, '').split

    words.each do |word|

      @word\_counts[word] ||= 0

      @word\_counts[word] += 1

    end

  end

  def output\_results

    sorted\_words = @word\_counts.sort\_by { |\_word, count| -count }

    puts "Top 10 Most Frequent Words:"

    sorted\_words[0..9].each do |word, count|

      puts "#{word}: #{count}"

    end

  end

end

# Example usage

filename = 'sample.txt'

word\_counter = WordCounter.new(filename)

word\_counter.analyze\_word\_usage

sample.txt

Mahendra Singh Dhoni (born 7 July 1981) is an Indian professional cricketer who plays as a right-handed batter and a wicket-keeper. Widely regarded as one of the most prolific wicket-keeper batsmen and captains, he represented the Indian cricket team and was the captain of the side in limited overs formats from 2007 to 2017 and in test cricket from 2008 to 2014.

Output

