<?php

class Calculation {

private $number;

private $maxValue;

private $iterations;

public function \_\_construct($number, $maxValue, $iterations) {

$this->number = $number;

$this->maxValue = $maxValue;

$this->iterations = $iterations;

}

public function getNumber() {

return $this->number;

}

public function getMaxValue() {

return $this->maxValue;

}

public function getIterations() {

return $this->iterations;

}

}

class CalculationManager {

public static function nextCalculation($x) {

return $x % 2 == 0 ? $x / 2 : 3 \* $x + 1;

}

public static function rangeCalculation($start, $end) {

$results = [];

for ($i = $start; $i <= $end; $i++) {

$number = $i;

$currentValue = $number;

$maxValue = $currentValue;

$iterations = 0;

while ($currentValue != 1) {

$currentValue = self::nextCalculation($currentValue);

$maxValue = max($maxValue, $currentValue);

$iterations++;

}

$results[] = new Calculation($number, $maxValue, $iterations);

}

return $results;

}

public static function maxIter($results) {

$maxIterations = 0;

$maxIterationsNumber = null;

foreach ($results as $result) {

if ($result->getIterations() > $maxIterations) {

$maxIterations = $result->getIterations();

$maxIterationsNumber = $result;

}

}

return $maxIterationsNumber;

}

public static function minIter($results) {

$minIterations = PHP\_INT\_MAX;

$minIterationsNumber = null;

foreach ($results as $result) {

if ($result->getIterations() < $minIterations) {

$minIterations = $result->getIterations();

$minIterationsNumber = $result;

}

}

return $minIterationsNumber;

}

}

// Example usage:

$results = CalculationManager::rangeCalculation(1, 100);

$maxIterationsNumber = CalculationManager::maxIter($results);

$minIterationsNumber = CalculationManager::minIter($results);

echo "Number with maximum iterations: " . $maxIterationsNumber->getNumber() . "\n";

echo "Number with minimum iterations: " . $minIterationsNumber->getNumber() . "\n";

?>