// Kelvin Kellner

// Mrs. Cooper

// 14 February 2019

// Unit One Quiz One (V2)

import java.util.\*; // import Java utility package

public class KelvinKellnerUnit1Quiz1 {

// Main Method: used for User Interface and Scanner interactions

public static void main(String[] args) {

Scanner scan = new Scanner(System.in); // new Scanner

// CREATE RANDOM INT ARRAY

// Prompt for values for random integer generator and store them accordingly

System.out.println("How many random integers would you like to create?");

int size = scan.nextInt();

System.out.println("What is the lowest integer that we can randomly generate?");

int min = scan.nextInt();

System.out.println("What is the highest integer that we can randomly generate?");

int max = scan.nextInt();

scan.nextLine(); // clear scanner

int[] nums = fillRandoms(size, min, max); // run the method to create and fill the array

// PRINT RANDOM INT ARRAY

// Print out the random integer array

System.out.println("\nThe numbers we generated were:");

for (int i = 0; i < nums.length; i++)

System.out.println(nums[i]);

// CALCULATE % DIVISIBLE

// Prompt the user to calculate the percentage of random numbers divisible by a certain value

System.out.println("\nWould you like to calculate the % of the random integers generated divisible by a certain value? (type y/n)");

String response = scan.nextLine();

if (response.equals("y") || response.equals("yes"))

{

System.out.println("\nOkay, great!\nWhat value would you like to divide by?");

double div = scan.nextDouble();

double perc = percDiv(nums, div); // calculate the %

System.out.println(perc + "% of randoms are divisble by " + div); // print result

}

else // If they say no to calculating the percentage

System.out.println("\nOkay, maybe next time :)");

// COMPARE STRING ARRAYS

String[] arrayOne = {"cat", "dog", "fish"}; // YOUR VALUES HERE!

String[] arrayTwo = {"cat", "dog", "fish"}; // YOUR VALUES HERE!

// Notify the user

System.out.println("\n\nWe will now compare two String arrays to see if the values are the same.");

System.out.println("Sorry, these values are hard-coded so you can't see the process :(\n");

boolean result = compareArrays(arrayOne, arrayTwo); // comparison time!

// Print out the result

if(result) // if the result is true

System.out.println("The String arrays are identical");

else // if the result is false

System.out.println("The String arrays are different");

scan.close(); // close Scanner, bye-bye program!

} // End Main Method

// Fill Randoms Method: Creates a new array of a given size and fills them within random numbers within a given range

// Parameters...

// - size (int): the # of numbers to generate (size of the array)

// - min (int): the lowest number possible to randomly generate

// - max (int): the highest number possible to randomly generate

// Returns: nums (int array): the final array of random integers

public static int[] fillRandoms(int size, int min, int max) {

Random random = new Random(); // create Random object

int[] nums = new int[size]; // create an empty array to store our values

for (int i = 0; i < nums.length; i++) // for each value in the array...

nums[i] = random.nextInt(max - min + 1) + min; // generate a random number within the given range of values

return nums; // return our filled array

} // End Fill Randoms Method

// Percentage Division Method: Calculates the % of values in an array divisible by a given value

// Parameters:

// - array (int array): the array to search through for division process

// - div (double): the number to divide by

// Returns: perc (double): the value of the percentage of numbers in the array divisible by a given value (rounded to the nearest 10th of a percent)

public static double percDiv(int[] array, double div)

{

double count = 0.0; // the # of array values divisible by our number

for(int i=0;i<array.length;i++)

{

if(((((double)(array[i]))/div)%1)==0.0) // if the remainder after dividing is equal to 0 (meaning the number divides correctly)...

count++; // add to the count of divisible numbers

}

double perc = Math.round((count/(double)array.length)\*1000.0)/10.0; // multiply the percentage to 100 to display as percentage, then round off all the digits past the tenths value

return perc; // return our percentage

} // End Percentage Division Method

// Compare Arrays Method: Compare two String arrays to see if they contain the same values

// Parameters:

// - one (String array): the first String array to compare

// - two (String array): the second String array to compare

// Returns: true - the arrays are identical, false - the arrays contain different values

public static boolean compareArrays(String[] one, String[] two)

{

if(one.length!=two.length) // if the arrays are of different lengths...

return false; // immediately return false, because they cannot be identical and have different lengths (yayyyy, we also prevent error messages this way!)

// If the arrays are the same length...

for(int i=0;i<one.length;i++) // check through each each value, and...

{

if(!one[i].equals(two[i])) // if (at any point) we find a value that does not exist within both arrays...

return false; // return false immediately because the arrays are not identical

}

return true; // if we make it through the whole array without returning false, then we can return true because the arrays are identical after all!!

} // End Compare Arrays Method

} // End Class