// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 6 Problem 6.24

// Date: 02/08/2023

// Language: Java

// File Name: Exercise6\_24.java

// Description: Perfect Numbers

// ------------------------------------------

import java.util.ArrayList;

import java.lang.StringBuilder;

public class Exercise6\_24 {

// C-like "Macro definition" equivalent size variable

private static final int size = 1000;

private static String isPerfect(int test) {

int sum = 0;

ArrayList<Integer> factors = new ArrayList<>();

// Find factors and add to list

for (int i = 1; i < test; i ++) {

if (test % i == 0) {

factors.add(i);

}

}

// IF factors are detected

if (factors.size() > 0) {

// Add them up

for (int i = 0; i < factors.size(); i++) {

sum += factors.get(i).intValue();

}

// IF perfect generate and return output. If not, return nothing

if (sum == test) {

StringBuilder output = new StringBuilder();

output.append(Integer.toString(test));

output.append(" - ");

output.append(factors.get(0).intValue());

for (int i = 1; i < factors.size(); i++) {

String s = " + " + factors.get(i).intValue();

output.append(s);

}

return output + "\n";

} else { return ""; }

}

// Default return

return "";

}

public static void main(String[] args) throws Exception {

System.out.printf("\nImperfect numbers ranging from 1 to %d\n\n", size);

// Find out if numbers are perfect from 0 to "size"

for (int i = 0; i < size; i++) {

System.out.print(isPerfect(i));

}

System.out.println();

}

}

// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 6 Problem 6.29

// Date: 02/08/2023

// Language: Java

// File Name: Exercise6\_29.java

// Description: Coin Tossing

// ------------------------------------------

import java.security.SecureRandom;

import java.util.Scanner;

public class Exercise6\_29 {

enum Face { HEADS, TAILS }

static int tailsCount;

static int headsCount;

static Face flip() {

// FLip the coin and return results

SecureRandom rand = new SecureRandom();

if (rand.nextInt(2) == 0) {

headsCount ++;

return Face.HEADS;

} else {

tailsCount ++;

return Face.TAILS;

}

}

public static void main(String[] args) throws Exception {

// Declare/initialize variables

int iterations;

Scanner input = new Scanner(System.in);

// User input for iteration limit

System.out.print("\n\nHow many times do you want to flip the coin? ");

iterations = input.nextInt();

// "FLip" for "iterations" iterations

for (int i = 0; i < iterations; i++) {

System.out.println("The result of the flip was " + flip());

}

// Format output

String output = "The coin was flipped " + iterations +

" times\nA result of heads was found " + headsCount +

" times\nA result of tails was found " + tailsCount +

" times\n\n";

input.close();

System.out.println(output);

}

}

// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 6 Problem 6.32

// Date: 02/08/2023

// Language: Java

// File Name: Exercise6\_32.java

// Description: Distance Between Points

// ------------------------------------------

import java.lang.Math;

import java.util.Scanner;

public class Exercise6\_32 {

static double getCoordinateDifference(double[] setOne, double [] setTwo) {

// Extract coordinates

double x1 = setOne[0];

double y1 = setOne[1];

double x2 = setTwo[0];

double y2 = setTwo[1];

// Return absolute value of sqrt((x2 - x2)^2 + (y2 - y1)^2)

return Math.abs(Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 - y1, 2)));

}

static double[] getCoordinatePair() {

// Initialize/declare variables for block

double[] coordinates = {0,0};

String var = "x";

Scanner input = new Scanner(System.in);

// Assign x to array index 0, y to array index 1

for (int i = 0; i < 2; i++) {

if (i == 1) { var = "y"; }

System.out.printf("Enter a(n) %s coordinate: ", var);

coordinates[i] = input.nextDouble();

}

// Return double of type array containing user created x and y coordinate

return coordinates;

}

public static void main(String[] args) throws Exception {

double setOne[] = getCoordinatePair();

double setTwo[] = getCoordinatePair();

String output = "\nThe distance is " + getCoordinateDifference(setOne, setTwo);

System.out.println(output);

}

}

Text

Description automatically generated

Figure : Exercise 6.24

Text

Description automatically generated

Figure : Exercise 6.29

Text

Description automatically generated

Figure : Exercise 6.32