/\*\* Demo on using Math.min( value1, value2 ) & Math.max( value1, value2 )

Read number of entries from user and gather that number of integers

and report on the maximum integer found. Assume all entries are positive \*/

import java.util.\*;

public class UsingMath {

public static void main(String[] args) {

int maximum = 0, entries, nextInt;

Scanner keyboard = new Scanner(System.in);

System.out.println("How many integers do you want to enter: ");

entries = keyboard.nextInt();

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

nextInt = keyboard.nextInt();

maximum = Math.max(nextInt, maximum);

}

System.out.println("The largest number entered was: " + maximum);

}

}

/\*\* Slide #8 demo showing a more EFFICIENT way to find/count primes\*/

public class Tester2 {

public static void main(String[] args) {

long t1 = 0, t2 = 0, t3 = 0; // Timer variables

t1 = System.currentTimeMillis();

printPrimes2(100000);

t2 = System.currentTimeMillis();

printPrimes1(100000);

t3 = System.currentTimeMillis();

System.out.println("My method took " + (t2 - t1) + " milliseconds");

System.out.println("Author's method took " + (t3 - t2) + " milliseconds");

}

// Prints all prime numbers up to the given max. (Author's method)

public static void printPrimes1(int max) {

int totPrimes = 0;

if (max >= 2) {

// System.out.print("2");

totPrimes++;

for (int i = 3; i <= max; i++) {

if (countFactors(i) == 2) {

// System.out.print(", " + i);

totPrimes++;

}

}

System.out.println("\nTotal Primes found: " + totPrimes);

}

}

// A far more efficient way to find/count primes

public static void printPrimes2(int max) {

int totPrimes = 0;

if (max >= 2) {

// System.out.print("2");

totPrimes++;

for (int i = 3; i <= max; i += 2) {

if (isPrime(i)) {

// System.out.print(", " + i);

totPrimes++;

}

}

System.out.println("\nTotal Primes found: " + totPrimes);

}

}

// Returns how many factors the given number has.

public static int countFactors(int number) {

int count = 0;

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

if (number % i == 0) {

count++; // i is a factor of number

}

}

return count;

}

public static boolean isPrime(int number) {

int count = 0;

for (int i = 2; i <= number/2; i++) {

if (number % i == 0) {

return false; // i is a factor of number

}

}

return true;

}

}

/\*\* Java program to simulate rolling dice until a '7' is seen \*/

import java.util.\*; // Needed for Random class

public class DiceRolls {

public static void main(String[] args) {

Random rand = new Random();

int die1 = rand.nextInt(6) + 1, die2 = rand.nextInt(6) + 1;

int roll = die1 + die2, rolls = 1;

System.out.println("Roll: " + die1 + " + " + die2 + " = " + roll);

while (roll != 7) {

die1 = rand.nextInt(6) + 1;

die2 = rand.nextInt(6) + 1;

roll = die1 + die2;

rolls++;

System.out.println("Roll: " + die1 + " + " + die2 + " = " + roll);

}

String rollStr = (rolls == 1)? " roll": " rolls";

System.out.println("It took " + rolls + rollStr + " to hit 7");

}

}

/\*\* Basic Calculator program; demo menu choices with conditional execution \*/

import java.util.\*;

public class Calculator {

public static void main(String[] args) {

Scanner keyboard = new Scanner(System.in);

int choice = 0;

double num1 = 0, num2 = 0;

boolean imaginary = false;

System.out.println("\t\t Calculator \n");

String menu = "\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\n5. SquareRoot;

menu += \n6. Quit\n\nChoice: ";

do {

System.out.print(menu);

choice = keyboard.nextInt();

switch (choice) {

case 1:

System.out.print("Enter 2 numbers seperated by a space to see their sum: ");

num1 = keyboard.nextInt();

num2 = keyboard.nextInt();

System.out.println(num1 + " + " + num2 + " = " + (num1 + num2));

break;

case 2:

System.out.print("Enter 2 numbers seperated by a space to see their difference: ");

num1 = keyboard.nextInt();

num2 = keyboard.nextInt();

System.out.println(num1 + " - " + num2 + " = " + (num1 - num2));

break;

case 3:

System.out.print("Enter 2 numbers seperated by a space to see their product: ");

num1 = keyboard.nextInt();

num2 = keyboard.nextInt();

System.out.println(num1 + " \* " + num2 + " = " + (num1 \* num2));

break;

case 4:

System.out.print("Enter 2 numbers seperated by a space to see their quotient: ");

num1 = keyboard.nextInt();

num2 = keyboard.nextInt();

if (num2 != 0) {

System.out.println(num1 + " / " + num2 + " = " + (num1 / num2));

} else {

System.out.println("Division by zero not allowed!!");

}

break;

case 5:

System.out.print("Enter a number to see its square root: ");

imaginary = false;

num1 = keyboard.nextInt();

if (num1 < 0)

imaginary = true;

num2 = Math.sqrt(Math.abs(num1));

System.out.print("The square root of of " + num1 + " is " + num2);

if (imaginary)

System.out.print("i");

System.out.println();

break;

case 6:

System.out.println("Quitting");

break;

default: System.out.println("Invalid entry!!!");

}

} while(choice != 6);

}

}