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

// Author: Lauren Escobedo

// Assignment: Chapter 4 Problem 4.22

// Date: 02/05/2023

// Language: Java

// File Name: Exercise1.java

// Description: Tabular Output

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

import java.lang.Math;

public class Exercise1 {

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

// This program is so simple comments aren't really needed...

System.out.printf("%-10s%-10s%-10s%-10s\n", "N", "10\*N", "100\*N", "1000\*N");

for (int i = 1; i <= 5; i++) {

System.out.printf("%-10d%-10d%-10d%-10d\n", i, i\*10, i\*100, i\*1000);

}

}

}

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

// Author: Lauren Escobedo

// Assignment: Chapter 4 Problem 4.26

// Date: 02/05/2023

// Language: Java

// File Name: Exercise3.java

// Description: What does the program print?

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

import java.lang.Math;

public class Exercise3 {

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

int row = 10;

// Iterate for 10 rows

while (row >= 1) {

int column = 1;

// iterate for 10 columns

while (column <= 10) {

// Output 1 or 0 depending on if the row is even or odd

// Textbook was cut off, so I subbed in 1 and 0

System.out.print(row %2 == 1 ? 0 : 1);

++ column;

}

-- row;

System.out.println();

}

}

}

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

// Author: Lauren Escobedo

// Assignment: Chapter 4 Problem 4.31

// Date: 02/05/2023

// Language: Java

// File Name: Exercise2.java

// Description: Palindromes

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

import java.lang.Math;

import java.util.Scanner;

public class Exercise2 {

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

// Get user input

Scanner input = new Scanner(System.in);

System.out.print("Enter an integer to test for palindrome property: ");

int test = input.nextInt();

input.close();

// Set loop variables

int initial = test;

int reversed = 0;

while(test > 0) {

// Reverse the number

reversed = reversed \* 10 + test % 10;

test /= 10;

}

// Output initial input and reversed

String output = "Initial = " + initial +

"\nReversed = " + reversed;

System.out.println(output);

// Output results

if (initial == reversed) { output = "Input was a palindrome"; }

else { output = "Input was not a palindrome"; }

System.out.println(output);

}

}

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

// Author: Lauren Escobedo

// Assignment: Chapter 5 Problem 5.20

// Date: 02/05/2023

// Language: Java

// File Name: Exercise5.java

// Description: Calculating Pi

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

import java.lang.Math;

public class Exercise5 {

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

// Variables for sequence calculations

double pi = 3.14159, k = 4, even = 3, odd = 5, constant = 4;

// Flag checks if sequence has summed to 3.14159 yet

boolean flag = false;

// Perform sequence calculation

for (int i = 2; i <= 200000; i++) {

if (i % 2 == 0) {

k -= constant/even;

even += 4;

}

if (i % 2 != 0) {

k += constant/odd;

odd += 4;

}

// If/when sequence equals 3.14159, output and flag it

if ((Math.floor(k \* 1e5) / 1e5) == pi && flag == false) {

System.out.printf("The sequence's sum began with %.5f on iteration %d\n", pi, i);

flag = true;

}

}

}

}

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

// Author: Lauren Escobedo

// Assignment: Chapter 5 Problem 5.21

// Date: 02/05/2023

// Language: Java

// File Name: Exercise4.java

// Description: Pythagorean Triples

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

import java.lang.Math;

public class Exercise4 {

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

// FLag variable tracks how many trianges have been found

int flag = 0;

for (int side\_one = 1; side\_one <= 500; side\_one++) {

// Iterate through side1 length

for (int side\_two = side\_one; side\_two <= 500; side\_two++) {

// Iterate though side2 length

for (int hypo = side\_two; hypo <= 500; hypo++) {

// Iterate through hypotenuse length

if (Math.pow(side\_one, 2) + Math.pow(side\_two, 2) == Math.pow(hypo, 2)) {

// When triangle is found, increment flag and output sides

flag ++;

String output = flag + ": " + side\_one + " " + side\_two + " " + hypo;

System.out.println(output);

}

}

}

}

}

}

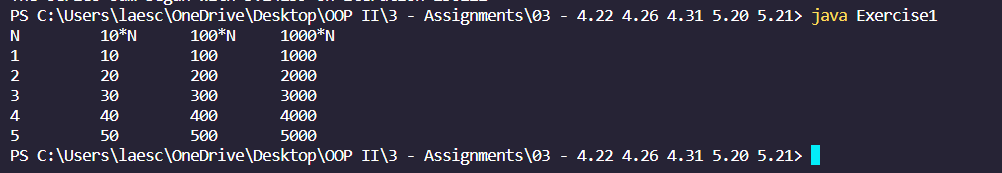


Figure : 4.22

Text

Description automatically generated

Figure : 4.26

A screenshot of a computer

Description automatically generated with medium confidence

Figure : 4.31

Text

Description automatically generated

Figure 5.20

Text

Description automatically generated

...

Text

Description automatically generated

Figure : 5.21