**CHALLENGE ACTIVITY. 7.6.1: Output of methods with loops.**

public class ComputationEngine {

public static int compute(int val1, int val2) {

int result;

int i;

result = 0;

for (i = 0; i < val1; ++i) {

result -= val2 + 3;

}

return result;

}

public static void main(String[] args) {

int value1;

int value2;

int computedValue;

value1 = 2;

value2 = 4;

computedValue = compute(value1, value2);

System.out.println(computedValue);

}

}

**Output: -14**

**CHALLENGE ACTIVITY. 7.6.2: Methods with loops.**

Define the method outputValues() that takes two integer parameters and outputs all integers starting with the first and ending with the second parameter in reverse order, each followed by a newline. The method does not return any value.

Ex: If the input is 1 3, then the output is:

3

2

1

**Answer:**

import java.util.Scanner;

public class ValuePrinter {

**public static void outputValues(int start, int end) {**

**int i;**

**for (i = end; i >= start; --i) {**

**System.out.println(i);**

**}**

**}**

public static void main(String[] args) {

Scanner scnr = new Scanner(System.in);

int num1;

int num2;

num1 = scnr.nextInt();

num2 = scnr.nextInt();

outputValues(num1, num2);

}

}

Define the method outputValue() that takes two integer parameters and outputs the product of all integers starting with the first and ending with the second parameter, followed by a newline. The method does not return any value.

Ex: If the input is 1 3, then the output is:

6

Note: Assume the first integer parameter is less than the second.

**Answer:**

import java.util.Scanner;

public class ValuePrinter {

public static void outputValue(int start, int end) {

int product = 1;

int i;

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

product = product \* i;

}

System.out.println(product);

}

public static void main(String[] args) {

Scanner scnr = new Scanner(System.in);

int input1;

int input2;

input1 = scnr.nextInt();

input2 = scnr.nextInt();

outputValue(input1, input2);

}

}

Define the method outputVal() that takes two integer parameters and outputs the product of all negative integers starting with the first and ending with the second parameter. If no negative integers exist, product is 1. End with a newline. The method does not return any value.

Ex: If the input is -4 7, then the output is:

24

Note: Negative numbers are less than 0.

**Answer:**

import java.util.Scanner;

public class ValuePrinter {

public static void outputVal(int start, int end) {

int product = 1;

int i;

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

if (i < 0) {

product = product \* i;

}

}

System.out.println(product);

}

public static void main(String[] args) {

Scanner scnr = new Scanner(System.in);

int numberA;

int numberB;

numberA = scnr.nextInt();

numberB = scnr.nextInt();

outputVal(numberA, numberB);

}

}

Define the method findSmallestVal() with a Scanner parameter that reads integers from input until a positive integer is read. The method returns the lowest of the integers read.

Ex: If the input is -5 -30 -85 90 -85, then the output is:

-85

Note: Positive numbers are greater than 0.

**Answer:**

import java.util.Scanner;

public class SmallestValFinder {

**public static int findSmallestVal(Scanner scnr) {**

**int inVal;**

**int minVal;**

**inVal = scnr.nextInt();**

**minVal = inVal;**

**while (inVal <= 0) {**

**if (inVal < minVal) {**

**minVal = inVal;**

**}**

**inVal = scnr.nextInt();**

**}**

**return minVal;**

**}**

public static void main(String[] args) {

Scanner scnr = new Scanner(System.in);

int minVal;

minVal = findSmallestVal(scnr);

System.out.println(minVal);

}

}

Define the method inspectVals() with a Scanner parameter that reads integers from input until -1 is read. The method returns true if all integers read before -1 are even. Otherwise, the function returns false.

Ex: If the input is 40 60 30 -1, then the output is:

All true

Note: Even numbers are divisible by 2.

**Answer:**

import java.util.Scanner;

**public class ValueInspector {**

**public static boolean inspectVals(Scanner scnr) {**

**int val;**

**val = scnr.nextInt();**

**while (val != -1) {**

**if (val % 2 != 0) {**

**return false;**

**}**

**val = scnr.nextInt();**

**}**

**return true;**

**}**

public static void main(String[] args) {

Scanner scnr = new Scanner(System.in);

boolean allEven;

allEven = inspectVals(scnr);

if (allEven) {

System.out.println("All true");

}

else {

System.out.println("Not all true");

}

}

}