
Lab Assignment 4

Nested Loops



CSE110: Programming Language I

	Lab Tasks	Home Tasks
Coding	3	5
Tracing	1	2

[You need to submit only the Home Tasks. Submit all the Homework Coding Tasks (Task 1 to 5) in the Google Form shared on buX. Submit all the Homework Tracing Tasks (Task 6 & 7) handwritten to your Lab Instructors before the next lab.]

LAB Tasks

(No Need to Submit)

1. Write a Java program to take a positive integer N (where $N > 0$) as user input and print the **first N prime numbers starting from 2**. Your code should check all the positive integers starting from 2 and determine whether they are prime or not until N prime numbers are found.

Sample Input 1:

5

Sample Output 1:

2

3

5

7

11

Sample Input 2:

7

Sample Output 2:

2

3

5

7

11

13

17

2. Write a Java code of a program that reads the value of N (where $N > 0$) from the user and calculates the value of y if the expression of y is as follows:

$$y = - (1) - (1 + 2) - (1 + 2 + 3) - \dots - (1 + 2 + 3 + \dots + N)$$

Sample Input:

The value of N : 2

Sample Output:

The value of y : -4

Sample Input:

The value of N : 4

Sample Output:

The value of y: -20

3. Write a Java program that will keep taking even positive integer numbers as inputs from the user and print the number of divisors of those numbers until it gets an odd number and then stops.

Sample Input & Output: (The purple numbers are input)

Enter Number: 44

44 has 6 divisors

Enter Number: 30

30 has 8 divisors

Enter Number: 8

8 has 4 divisors

Enter Number: 4

4 has 3 divisors

Enter Number: 6

6 has 4 divisors

Enter Number: 20

20 has 6 divisors

Enter Number: 24

24 has 8 divisors

Enter Number: 5

4. Trace the following code, create a tracing table and write the outputs.

1	<code>public class T1{</code>
2	<code> public static void main(String args[]){</code>
3	<code> int x = 0, y = 0;</code>
4	<code> int sum = 0;</code>
5	<code> while (x < 4){</code>
6	<code> y = x - 3;</code>
7	<code> while (y < 3){</code>
8	<code> sum = (sum % 3) + x - y * 3 ;</code>
9	<code> System.out.println(sum);</code>
10	<code> y = y + 1;</code>
11	<code> }</code>
12	<code> if (x > 5){</code>
13	<code> x++;</code>
14	<code> }</code>
15	<code> else{</code>
16	<code> x += 2;</code>
17	<code> }</code>
18	<code> }</code>
19	<code> }</code>
20	<code>}</code>

Home Tasks

1. Read an integer N that is the number of test cases that follow. Each test case contains two integers X and Y. Print one output line for each test case that the sum of Y odd numbers from X including it if is the case. For example:

For the input 4 5, the output must be 45, that is: $5 + 7 + 9 + 11 + 13$

For the input 7 4, the output must be 40, that is: $7 + 9 + 11 + 13$

Sample Input	Sample Output
2 4 3 11 2	21 24
Explanation: Here, the 2 means there are two test cases. For each test case you have to take two inputs (X, Y) and print the sum of Y odd numbers starting from X.	

2. Take the length and width of a **rectangle** from the user and create the rectangle according to the output below. Your output should match the specified output.

[illegible]

3. Write a Java program that asks the user to enter the starting and the ending number of a range. Then the program finds out all the prime numbers between that range.

Sample Input	Sample Output
Start: 5 End: 20	5 7 11 13 17 19

4. Write a java program that will take the number of customers as input from the user (n) and then take the number of items (k) and finally take k inputs which are the price of each product. Your program should print the average price of the items for that customer.

Sample Input	Sample Output
Number of Customer: 2 Item Number for Customer-1: 3 Item-1: 20 Item-2: 30 Item-3: 70 Item Number for Customer-2: 5 Item-1: 35 Item-2: 120 Item-3: 75 Item-4: 13 Item-5: 33	Average for Customer-1: 40 Average for Customer-2: 55.2

5. Write a Java program that will ask for a range (a starting number and an ending number) from the user and print all the Armstrong numbers between that range.

[Armstrong Number: An Armstrong number is a number whose sum of digits raised to the power the number of digits equals to that number.

For example, 371 is an Armstrong number because $3^3 + 7^3 + 1^3 = 371$, here the total number of digits in 371 is 3]

Sample Input 1:

Start: 300

End: 500

Sample Output 1:

Armstrong numbers:

370

371

407

Sample Input 2:

Start: 100

End: 200

Sample Output 2:

Armstrong numbers:

153

6. Trace the following code, create a tracing table and write the outputs.

1	<code>public class Tracing{</code>
2	<code> public static void main(String[] args) {</code>
3	<code> int sum = 0, r = 0, i = 10, j = -5000;</code>
4	<code> while (i < 10000) {</code>
5	<code> r = (i%2) + 1;</code>
6	<code> j = i + 2;</code>
7	<code> while (j < i){</code>
8	<code> r = 2-4+6-8+10-12;</code>
9	<code> j++;</code>
10	<code> }</code>
11	<code> sum = i % r + (i * r) - j + 5;</code>
12	<code> System.out.println(sum);</code>
13	<code> j--;</code>
14	<code> if (i > 18) {</code>
15	<code> sum += (++r) + i/3 + r + 12;</code>
16	<code> break;</code>
17	<code> }</code>
18	<code> else{</code>
19	<code> i += 3;</code>
20	<code> }</code>
21	<code> }</code>
22	<code> System.out.println(sum);</code>
23	<code> }</code>
24	<code>}</code>

7. Trace the following code, create a tracing table and write the outputs.

1	<code>public class T3</code>
2	<code>{</code>
3	<code> public static void main(String args[])</code>
4	<code> {</code>
5	<code> int x = 0, y = 0;</code>
6	<code> int sum = 0;</code>
7	<code> while (x < 10){</code>
8	<code> y = x - 3;</code>
9	<code> y = 40;</code>
10	<code> while (y > 22){</code>
11	<code> if ((sum > 30) && (sum < 40)){</code>
12	<code> sum = sum + x * 2 ;</code>
13	<code> }</code>
14	<code> else if ((sum > 40) && (sum < 50)){</code>
15	<code> sum = sum + x * 3;</code>
16	<code> }</code>
17	<code> else {</code>
18	<code> sum = sum + 23;</code>
19	<code> }</code>
20	<code> System.out.println(sum) ;</code>
21	<code> y = y - 10;</code>
22	<code> }</code>
23	<code> x += 2;</code>
24	<code> }</code>
25	<code> }</code>
26	<code>}</code>