LAB ASSIGNMENT # 04

Nested Loop

BRAC UNIVERSITY

Inspiring Excellence

CSE110 | Programming Language I

	LAB TASKS	HOME TASKS
CODING	04	04
TRACING	01	02

NOTE: You need to submit only the Home Tasks. Submit all the Flowchart or Tracing tasks hand drawn or handwritten respectively to your Lab Instructors before the next lab. Submit all the Homework Coding Tasks in the Google Form shared on buX.

LAB TASKS

[NO NEED TO SUBMIT]

Question: 1

In a virtual arena, there are N warriors numbered from 1 to N. Every warrior fights every other warrior exactly once. A warrior wins if their strength (their number) is divisible by the opponent's strength. Your job is to write a java code that calculates the total number of duels won in the entire tournament.

Sample Input	Sample Output
5	2 beats 1 3 beats 1 4 beats 1 4 beats 2 5 beats 1 Total wins: 5
8	2 beats 1 3 beats 1 4 beats 1 4 beats 2 5 beats 1 6 beats 1 6 beats 2 6 beats 3 7 beats 1 8 beats 1 8 beats 2 8 beats 4 Total wins: 12

Question: 2

Students at Hero Academia must complete missions and earn points in order to pass a grade. The homeroom teacher of class 1A wants to identify the top performer and give them a badge of honour. He provides you with the number of students (S), the number of missions (M) these students completed and for each student then gives you how many points they collected on each of these missions.

Write the java code to calculate each student's average points and help the teacher find the top performer. [N.B: All students receive points for all (M number of) missions.]

Sample Input	Sample Output
Students (S): 3 Missions (M): 3 Scores of Student 1: 10 9 8 Scores of Student 2: 9 10 10 Scores of Student 3: 7 8 9	Average point of Student 1: 9.0 Average point of Student 2: 9.67 Average point of Student 3: 8.0 Student 2 receives the badge of honour!!

Question: 3

You are testing SecureVault's safe locks. The program should ask for how many pairs (N) of codes to test, then for each pair, take two positive numbers and find their GCD using the Euclidean method. If any number is 0 or negative, print 0.

*[**Hint: (Euclidean method)** Two positive integers a and b. Assume a is greater than b, then divide a by b and find the remainder r.This can be expressed as a = quotient * b + r.*

If r = 0, b is the GCD of a and b and the loop terminated.

If r != 0, replace a with b and b with r and repeat the previous step.]

Example:

Steps	*a*	*b*	*a%b*	*Actions*
1	*48*	*18*	*12*	*a←18, b←12*
2	*18*	*12*	*6*	*a←12, b←6*

3	*12*	*6*	*0*	*Stop→GCD = 6*

Input	Output
Enter number of pairs: 2 Pair 1: Enter first number: 40 Enter second number: 50 Pair 2: Enter first number: 0 Enter second number: 32	Pair 1: GCD = 10 Pair 2: GCD = 0

Question: 4
At Nimbus Tech, the HR analytics team is studying the “Stress Level” of a particular employee over “W” number of weeks. Each week the employee takes on a task more difficult than the previous week and adds on to his workload. This each week’s workload up to week “W”, is subtracted to find his Total Stress level.

For week 1, the workload is 1.
For week 2, it’s the sum of workload of previous week and this week’s new project (1+2).
For week 3, it’s (1+2+3). And so,
For week W, it’s (1+2+3+...+W)

The “Stress Level” also written as STRL is the result of subtracting all the weeks’ workload.

$$\mathbf{STRL = -(1) - (1+2) - (1+2+3) - \ldots - (1+2+3+\ldots+W)}$$

Write a java code for the HR team to find the STRL of the employee when given “W” or number of weeks.

Sample Input	**Sample Output**
Enter number of Weeks, W: 4	STRL= -20
Explanation: Since W = 4, STRL = - (1) - (1+2) - (1+2+3) - (1+2+3+4) = - 1 - 3 - 6 - 10 = -20	

Question: 5

```
public class T1{
   public static void main(String args[]){
      int x = 0, y = 0;
      int sum = 0;
      while (x < 4){
        y = x - 3;
        while (y < 3){
          sum = (sum % 3) + x - y * 3 ;
          System.out.println(sum);
          y = y + 1;
        }
        if (x > 5){
            x++;
        }
        else{
            x += 2;
        }
      }
   }
}
```

HOME TASKS

Question: 1

NarcissiTech is a luxury telecom company that sells "Vanity Phone Numbers", special numbers that customers love because they are mathematically elegant. One of their flagship products is "Armstrong Vanity Numbers", where the number itself equals the sum of its digits each raised to the power of the total number of digits.

You need to write a Java program that asks the user to input a starting number and an ending number. The program must find and display all Armstrong numbers within this range (inclusive).

[An Armstrong number is a number whose digits' values, each raised to the power of the total number of digits, add up to the number itself.

For example:

- *371 has 3 digits*
- *So we calculate:*
  $3^3 + 7^3 + 1^3 = 27 + 343 + 1 = 371$
- *Since the sum equals the original number, 371 is an Armstrong number.]*

Question: 2

At Codewarts University of Magic and Programming, every student is assigned an unique numeric ID when they enroll. Professor Dumbledore, the head of the Department of Logical Arts, believes that students whose ID numbers are powers of 2 possess extraordinary logical abilities. Such students are said to have “Lucky IDs.”

Your task is to help the university identify which students have Lucky IDs. You will first take an user input of the number of test cases (N) representing how many students' IDs need to be checked. For each student, take input of their ID number and determine whether it is a power of two. Display "Lucky ID" if the student's ID is a power of two, otherwise display "Not Lucky".

Sample Input	Sample Output
Enter number of students to check: 5 Enter student ID: 1 Enter student ID: 2 Enter student ID: 3 Enter student ID: 8	Lucky ID Lucky ID Not Lucky Lucky ID Not Lucky

Enter student ID: 10	
Explanation: Student ID: 1 = 2^0 Student ID: 2 = 2^1 Student ID: 3 = Not a power of 2. Student ID: 8 = 2^3 Student ID: 10 = Not a power of 2.	

Question: 3

At *ShopSmart Mart*, the owner keeps track of daily sales of three products for several days. At the end of each day, the shop calculates the total sales. To comply with local regulations, the shop applies taxes on total sales:

- If the total sales for the day equal to or greater than 100, a 2% tax is added.
- If the total sales equal to or greater than 200, a 5% tax is applied instead.
- If the total sales equal to or greater than 500, a 10% tax is applied instead.

The owner wants your help to display the sales table for each day, showing the total sales including any applicable tax.

You need to write a Java program that asks the user to input the number of days (N). Then, for each day, input the sales of the 3 products. For each day, calculate the total sales, apply the correct tax based on the total, and display the total sales including tax.

Sample Input	**Sample Output**
Enter number of days: 4 Enter sales for Day 1: 30 40 50 Enter sales for Day 2: 40 50 60 Enter sales for Day 3: 100 80 50 Enter sales for Day 4: 200	Day 1: Total Sales with Tax: 122.4 Day 2: Total Sales with Tax: 153.0 Day 3: Total Sales with Tax: 241.5 Day 4: Total Sales with Tax: 583.0

<table>
<tr><td>150
180</td><td></td></tr>
<tr><td colspan="2"><b>Explanation:</b>
Sales for Day 1: 30+40+50 = 120
Total sales= 120 + (2% of 120) = 122.4

Sales for Day 2: 40+50+60 = 150
Total sales= 150 + (2% of 150) = 153.0

Sales for Day 3: 100+80+50 = 230
Total sales= 230 + (5% of 230) = 241.5

Sales for Day 4: 200+150+180 = 530
Total sales= 530 + (10% of 530) = 583.0</td></tr>
</table>

Question: 4

GymFit Elite uses an AI fitness system to calculate effective calorie burn per exercise with advanced rules. Your task is to write a java programme to calculate the average of earned calories. For each member, input the number of exercises and their raw calories. Each day a member can do the highest 3 exercises. If the calorie is greater than 350 then add 50% bonus, add 10% penalty to all if every exercise is less than 200, and give 50 calories flat bonus per exercise if the raw average is greater than 400. Now calculate the total calories earned and print the average calories earned a day.

Input	Output
Number of Members: 2 Exercises for Member-1: 3 Exercise-1: 300 Exercise-2: 400 Exercise-3: 800 Exercises for Member-2: 4 Exercise-1: 150 Exercise-2: 100 Exercise-3: 80	Average calories earned per day for Member-1: 750.000 (Can’t do more than 3 exercise) Average calories earned per day for Member-2: 99.000
Explanation: For first member (1), number of exercise is 3 Calorie for 3 exercises are 300, 400 and 800 So, rawSum = 300+400+800 = 1500 As calorie burn for exercise 2 and 3 is more than 350 so 50% bonus is added to them For 400, 200 will be added and for 800, 400 will be added, so total = 300 + 200 + 400 + 800 + 400 = 2100 Average of rawSum is 1500 / 3 = 500 The average of rawSum is greater than 400, therefore 50 bonus will be added for all three exercises. Therefore total is 2100 + 50 + 50 + 50 = 2250 Finally total average is: 2250 / 3 = 750 For second member (2), number of exercise is 4 which is greater than 3 so member need to perform maximum 3 exercises Calorie for 3 exercises are 150, 100 and 80 So, rawSum = 150+100+80 = 330 So, rawAvg = 330 / 3 = 110 As calorie burn for exercise 1, 2 and 3 is not more than 200, 10% penalty will be given 110 - 10% = 99 The average of rawSum is not greater than 400, therefore 50 bonus will not	

be added for all three exercises.

Question: 5

```
public class T3
{
  public static void main(String args[])
  {
      int x = 0, y = 0;
      int sum = 0;
      while (x < 10){
        y = x - 3;
        y = 40;
        while (y > 22){
          if ((sum > 30) && (sum < 40)){
              sum = sum + x * 2 ;
         }
          else if ((sum > 40) && (sum < 50)){
              sum = sum + x * 3;
         }
          else {
              sum = sum + 23;
         }
          System.out.println(sum);
          y = y - 10;
        }
        x += 2;
      }
  }
}
```

Question: 6

```
public class Tracing{
   public static void main(String[] args) {
       int sum = 0, r = 0, i = 10, j = -5000;
       while (i < 10000) {
           r = (i%2) + 1;
           j = i + 2;
           while (j < i){
               r = 2-4+6-8+10-12;
               j++;
           }
           sum = i % r + (i * r) - j + 5;
           System.out.println(sum);
           j--;
           if (i > 18) {
               sum += (++r) + i/3 + r + 12;
               break;
           }
           else{
               i += 3;
           }
       }
       System.out.println(sum);
   }
}
```

UNGRADED TASKS
(For Practice)

Questions: 1
In the ancient city of Luminalis, the Grand Scrollkeeper guards the Scroll of Know-it-all, a relic that grants you access to all knowledge in existence. To get the scroll you must pass the Scrollkeeper's test to find the sum of "unusual numbers". Unusual numbers are those that are not divisible by any number smaller than themselves, other than 1.

The scrollkeeper hands you two numbers: X (A starting point) and Y (The number of unusual numbers you must find). Write a java code to find the answer to the scrollkeeper's question.

Sample Input	Sample Output
X: 4 Y: 5	Sum: 53
Explanation: *5 unusual numbers starting from 4 are: 5, 7, 11, 13, 17* *Their sum: 5 + 7 + 11 + 13 + 17 = 53*	

Question: 2
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.	

Question: 3

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
```

Question: 4

Read one integer P (12 ≤ P ≤ 1000). Find all integer triples (a, b, c) with 1 ≤ a < b < c such that a + b + c = P and a*a + b*b = c*c. Print each triple on its own line as three numbers separated by single spaces, then print the total number of such triples on the last line. If there are no solutions, print 0 only.

Sample Input	Sample Output
60	10 24 26 15 20 25 Total- 2
120	20 48 52 24 45 51 30 40 50 Total- 120