Problem Statements

Premanand S

Assistant Professor
School of Electronics Engineering
Vellore Institute of Technology
Chennai Campus

premanand.s@vit.ac.in

February 2, 2026

Practice Questions 1

Rahul books a resort stay. Room charges per day: Economy – Rs:800, Premium – Rs: 1200, Luxury – Rs:1600. Optional services: Meals – Rs:250/day, Transport – Rs:150/day. If the stay exceeds 7 days, apply a 12% discount on the total bill. Write a program to calculate the final cost.

Resort Stay Bill Calculator – Part 1

```
import java.util.*;
public class Main {
 public static void main(String[] args) {
  Scanner sc = new Scanner(System.in);
  int roomType = sc.nextInt();
  int stayDays = sc.nextInt();
  int mealsChoice = sc.nextInt();
  int transportChoice = sc.nextInt();
  int roomCostPerDay = 0;

  if (roomType == 1)
         roomCostPerDay = 800;
  else if (roomType == 2)
      roomCostPerDay = 1200;
  else
      roomCostPerDay = 1600;
  int totalBill = roomCostPerDay * stayDays;
```

Resort Stay Bill Calculator – Part 2

```
  if (mealsChoice == 1)
      totalBill = totalBill + (250 * stayDays);

  if (transportChoice == 1)
      totalBill = totalBill + (150 * stayDays);

  if (stayDays > 7)
      totalBill = totalBill - (totalBill * 12 / 100);

  System.out.println(totalBill);
 }
}
```

Practice Questions 2

Find the sum of all two-digit numbers between L and R whose sum of digits is a composite number.

Sum of Two-Digit Numbers (Digit Sum is Composite) – Part 1

```
import java.util.*;
public class Main {
 static boolean isComposite(int num) {
  if (num <= 1)
  return false;
  for (int i = 2; i <= num / 2; i++) {
   if (num % i == 0)
       return true;
  }
  return false;
 }
 public static void main(String[] args) {
  Scanner sc = new Scanner(System.in);
  int lowerLimit = sc.nextInt();
  int upperLimit = sc.nextInt();
  int totalSum = 0;
```

Sum of Two-Digit Numbers (Digit Sum is Composite) – Part 2

```
  for (int number = lowerLimit; number <= upperLimit; number++) {

   int tensDigit = number / 10;
   int onesDigit = number % 10;

   int digitSum = tensDigit + onesDigit;

   if (isComposite(digitSum)) {
    totalSum = totalSum + number;
   }
  }

  System.out.println(totalSum);
 }
}
```

Practice Questions 3

Count how many three-digit numbers in a given range are formed using only the digits $\{3, 6, 9\}$.

Count Valid 3-Digit Numbers (Digits {3,6,9}) – Part 1

```
import java.util.*;
public class Main {
 static boolean isValidNumber(int number) {
  while (number > 0) {
   int digit = number % 10;

   if (!(digit == 3 || digit == 6 || digit == 9)) {
    return false;
   }
   number = number / 10;
  }
  return true;
 }
 public static void main(String[] args) {
  Scanner sc = new Scanner(System.in);
  int startRange = sc.nextInt();
  int endRange = sc.nextInt();
  int validCount = 0;
```

Count Valid 3-Digit Numbers (Digits {3,6,9}) – Part 2

```
  for (int number = startRange; number <= endRange; number++) {
   if (number >= 100 && number <= 999) {
    if (isValidNumber(number)) {
     validCount++;
    }
   }
  }

  if (validCount == 0)
  System.out.println("No valid numbers");
  else
  System.out.println("Count of valid numbers: " + validCount);
 }
}
```

Practice Questions 4

Find all two-digit numbers in a range where (square of the first digit square of the second digit) is a perfect cube.

Two-Digit Numbers with Perfect Cube Difference – Part 1

```
import java.util.*;
public class Main {
 static boolean isPerfectCube(int value) {
  if (value <= 0)
      return false;
  int cubeRoot = (int) Math.round(Math.cbrt(value));
  return cubeRoot * cubeRoot * cubeRoot == value;
 }
 public static void main(String[] args) {
  Scanner sc = new Scanner(System.in);
  int startRange = sc.nextInt();
  int endRange = sc.nextInt();
  boolean found = false;
  for (int number = startRange; number <= endRange; number++) {
```

Two-Digit Numbers with Perfect Cube Difference – Part 2

```
  if (number >= 10 && number <= 99) {

   int firstDigit = number / 10;
   int secondDigit = number % 10;

   int difference = (firstDigit * firstDigit)
   - (secondDigit * secondDigit);

   if (isPerfectCube(difference)) {
    System.out.print(number + " ");
    found = true;
   }
  }
 }
 if (!found)
 System.out.println("No valid numbers found in the given

 range.");
}
```

Practice Questions 5

Print all numbers between two limits where every non-zero digit divides the number exactly. Also print their average.

Valid Numbers and Average (Digit Divisibility) – Part 1

```
import java.util.*;
public class Main {
 static boolean isValidNumber(int number) {
  int temp = number;
  while (temp > 0) {
   int digit = temp % 10;
   if (digit == 0 || number % digit != 0) {
    return false;
   }
   temp = temp / 10;
  }
  return true;
 }
 public static void main(String[] args) {
  Scanner sc = new Scanner(System.in);
  int startRange = sc.nextInt();
  int endRange = sc.nextInt();
  int sum = 0;
  int count = 0;
```

Valid Numbers and Average (Digit Divisibility) – Part 2

```
  for (int number = startRange; number <= endRange; number++) {
   if (isValidNumber(number)) {
    System.out.print(number + " ");
    sum = sum + number;
    count++;
   }
  }
  if (count == 0) {
   System.out.println("No valid numbers");
  } else {
   int average = sum / count;
   System.out.println();
   System.out.println("Average: " + average);
  }
 }
}
```

Practice Questions 6

From a range of two-digit numbers, print all pairs (i, j) such that $|i - j|$ is a perfect cube.

Pairs with Perfect Cube Difference – Part 1

```
import java.util.*;
public class Main {
 static boolean isPerfectCube(int value) {
  if (value < 0)
  value = -value;
  int cubeRoot = (int) Math.round(Math.cbrt(value));
  return cubeRoot * cubeRoot * cubeRoot == value;
 }

 public static void main(String[] args) {
  Scanner sc = new Scanner(System.in);

  int startRange = sc.nextInt();
  int endRange = sc.nextInt();
```

Pairs with Perfect Cube Difference – Part 2

```
for (int firstNumber = startRange; firstNumber <= endRange;
firstNumber++) {

 if (firstNumber < 10 || firstNumber > 99)
 continue;

 for (int secondNumber = firstNumber + 1; secondNumber <=
 endRange; secondNumber++) {

  if (secondNumber < 10 || secondNumber > 99)
  continue;

  int difference = secondNumber - firstNumber;
```

Pairs with Perfect Cube Difference – Part 3

```
        if (isPerfectCube(difference)) {
         System.out.println("(" + firstNumber + ", "
         + secondNumber + ") Difference: " + difference);
        }
       }
      }
     }
    }
```

Practice Questions 7

Find all pairs (x, y) in a range such that $(x - y)$ is divisible by $(x + y)$.

Practice Questions 8

Print all three-digit numbers in a range where the sum of the first and last digits is 10.

Practice Questions 9

Count how many numbers between 1 and N have digits in strictly descending order.

Practice Questions 10

Find all two-digit numbers in a range that are divisible by both digits plus one. Print their sum.

Practice Questions 11

Reverse the elements of an array in groups of size k and print the product of elements in each group.

Practice Questions 12

Replace every number in an array with the closest even number. If there is a tie, choose the smaller one.

Practice Questions 13

From an array, print the element that appears more than $n/3$ times. If no such element exists, print an appropriate message.

Practice Questions 14

Given an array and a window size k, print the maximum element in every sliding window.

Practice Questions 15

Create a new array where each element is the sum of all elements except the one at that index.

Practice Questions 16

Given a matrix, count how many even numbers are present in each row.

Practice Questions 17

Print the elements of a matrix in spiral order starting from the top-right corner in clockwise direction.

Practice Questions 18

Replace every element in each column of a matrix with the sum of that column.

Practice Questions 19

Reverse each column of a matrix and find the absolute difference between the sum of the first row and the last row.

Practice Questions 20

Given a matrix, swap any two specified columns and print the updated matrix.

Thank You!

Stay Connected

Premanand S

Email:	premanand.s@vit.ac.in
Phone:	+91-7358679961
LinkedIn:	linkedin.com/in/premsanand
Instagram:	instagram.com/premsanand
WhatsApp Channel:	anandsDataX
Google Scholar:	Google Scholar Profile
GitHub:	github.com/anandprems