Level 3 Practice Programs

1. **Create a program to find the shortest, tallest, and mean height of players present in a football team.**

```java
import java.util.Arrays;
import java.util.Scanner;
public class Football {

    public static void main(String[] args) {
        int[] heights = new int[11];
        Scanner input = new Scanner(System.in);

        //User input
        System.out.println("Enter the heights (in cm) of 11 football players:");
        for (int i = 0; i < heights.length; i++) {
            System.out.print("Player " + (i + 1) + ": ");
            heights[i] = input.nextInt();
        }

        //Print results
        System.out.println("\n");
        System.out.println("Shortest Height: " + findShortest(heights) + " cm");
        System.out.println("Tallest Height: " + findTallest(heights) + " cm");
        System.out.println("Mean Height: " + findMean(heights) + " cm");
    }

    //Method to find shortest
    public static int findShortest(int[] arr) {
        int min = arr[0];
        for (int heights : arr) {
            if (heights < min) min = heights;
        }
        return min;
    }

    //Method to find tallest
    public static int findTallest(int[] arr) {
        int max = arr[0];
        for (int heights : arr) {
            if (heights > max) max = heights;
        }
        return max;
    }

    //Method to find sum for mean
    public static int findSum(int[] arr) {
        int sum = 0;
        for (int heights : arr) {
            sum += heights;
        }
        return sum;
    }

    //Method to find mean
    public static int findMean(int[] arr) {
        return findSum(arr) / arr.length;
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>javac Football.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java Football
Enter the heights (in cm) of 11 football players:
Player 1: 162
Player 2: 152
Player 3: 169
Player 4: 171
Player 5: 159
Player 6: 154
Player 7: 142
Player 8: 166
Player 9: 174
Player 10: 170
Player 11: 168


Shortest Height: 142 cm
Tallest Height: 174 cm
Mean Height: 162 cm
```

2. **Extend or Create a *NumberChecker* utility class and perform following task. Call from main() method the different methods and display results. Make sure all are static methods.**

```java
import java.util.Arrays;
import java.util.Scanner;
public class NumberChecker {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = input.nextInt();

        int[] digits = getDigits(number);
        int count = digits.length;

        System.out.println("\n--- Results ---");
        System.out.println("Count of Digits: " + count);
        System.out.println("Is Duck Number: " + isDuckNumber(digits));
        System.out.println("Is Armstrong Number: " + isArmstrong(number, digits));
        findTwoLargest(digits);
        findTwoSmallest(digits);
    }

    // Get digits of the number
    public static int[] getDigits(int number) {
        String numStr = String.valueOf(number);
        int[] arr = new int[numStr.length()];
        for (int i = 0; i < numStr.length(); i++) {
            arr[i] = numStr.charAt(i) - '0';
        }
        return arr;
    }

    // Check if it's a duck number (has a 0 but not at the start)
    public static boolean isDuckNumber(int[] digits) {
        for (int i = 1; i < digits.length; i++) {
            if (digits[i] == 0) return true;
        }
        return false;
    }

    // Check Armstrong number
    public static boolean isArmstrong(int number, int[] digits) {
        int sum = 0;
        int power = digits.length;
        for (int d : digits) {
            sum += Math.pow(d, power);
        }
        return sum == number;
    }
```

```java
    // Find and print largest and second largest digits
    public static void findTwoLargest(int[] arr) {
        int max1 = Integer.MIN_VALUE;
        int max2 = Integer.MIN_VALUE;
        for (int n : arr) {
            if (n > max1) {
                max2 = max1;
                max1 = n;
            } else if (n > max2 && n != max1) {
                max2 = n;
            }
        }
        System.out.println("Largest Digit: " + max1);
        System.out.println("Second Largest Digit: " + (max2 == Integer.MIN_VALUE ? "N/A" : max2));
    }

    // Find and print smallest and second smallest digits
    public static void findTwoSmallest(int[] arr) {
        int min1 = Integer.MAX_VALUE;
        int min2 = Integer.MAX_VALUE;
        for (int n : arr) {
            if (n < min1) {
                min2 = min1;
                min1 = n;
            } else if (n < min2 && n != min1) {
                min2 = n;
            }
        }
        System.out.println("Smallest Digit: " + min1);
        System.out.println("Second Smallest Digit: " + (min2 == Integer.MAX_VALUE ? "N/A" : min2));
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java NumberChecker
Enter a number: 68

--- Results ---
Count of Digits: 2
Is Duck Number: false
Is Armstrong Number: false
Largest Digit: 8
Second Largest Digit: 6
Smallest Digit: 6
Second Smallest Digit: 8
```

3. **Extend or Create a *NumberChecker* utility class and perform following task. Call from main() method the different methods and display results. Make sure all are static methods.**

```java
import java.util.Arrays;
import java.util.Scanner;
public class NumberChecker2 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = input.nextInt();

        int count = countDigits(number);
        int[] digits = getDigitsArray(number);

        System.out.println("\n--- Results ---");
        System.out.println("Count of Digits: " + count);
        System.out.println("Sum of Digits: " + sumOfDigits(digits));
        System.out.println("Sum of Squares of Digits: " + sumOfSquares(digits));
        System.out.println("Is Harshad Number: " + isHarshad(number, digits));
        printDigitFrequency(digits);
    }

    // Method to count digits
    public static int countDigits(int number) {
        return String.valueOf(number).length();
    }

    // Method to get digits array
    public static int[] getDigitsArray(int number) {
        String str = String.valueOf(number);
        int[] arr = new int[str.length()];
        for (int i = 0; i < str.length(); i++) {
            arr[i] = str.charAt(i) - '0';
        }
        return arr;
    }

    // Method to find sum of digits
    public static int sumOfDigits(int[] digits) {
        int sum = 0;
        for (int d : digits) {
            sum += d;
        }
        return sum;
    }
```

```java
    // Method to find sum of squares of digits
    public static int sumOfSquares(int[] digits) {
        int sum = 0;
        for (int d : digits) {
            sum += Math.pow(d, 2);
        }
        return sum;
    }

    // Method to check if the number is a Harshad number
    public static boolean isHarshad(int number, int[] digits) {
        int sum = sumOfDigits(digits);
        return number % sum == 0;
    }

    // Method to find digit frequencies
    public static void printDigitFrequency(int[] digits) {
        int[][] freq = new int[10][2]; // [digit][frequency]

        // Initialize digit values
        for (int i = 0; i < 10; i++) {
            freq[i][0] = i;
            freq[i][1] = 0;
        }

        // Count frequencies
        for (int d : digits) {
            freq[d][1]++;
        }

        System.out.println("Digit Frequencies:");
        for (int i = 0; i < 10; i++) {
            if (freq[i][1] > 0) {
                System.out.println("Digit " + freq[i][0] + ": " + freq[i][1] + " time(s)");
            }
        }
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java NumberChecker2
Enter a number: 74

--- Results ---
Count of Digits: 2
Sum of Digits: 11
Sum of Squares of Digits: 65
Is Harshad Number: false
Digit Frequencies:
Digit 4: 1 time(s)
Digit 7: 1 time(s)
```

4. **Extend or Create a *NumberChecker* utility class and perform following task. Call from main() method the different methods and display results. Make sure all are static methods.**

```java
import java.util.Arrays;
import java.util.Scanner;
public class NumberChecker3 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = input.nextInt();

        int[] digits = getDigits(number);
        int[] reversed = reverseArray(digits);

        System.out.println("\n--- Results ---");
        System.out.println("Count of Digits: " + digits.length);
        System.out.println("Is Palindrome: " + isPalindrome(digits, reversed));
        System.out.println("Is Duck Number: " + isDuckNumber(digits));
    }

    // Get digits from number
    public static int[] getDigits(int num) {
        String str = String.valueOf(num);
        int[] arr = new int[str.length()];
        for (int i = 0; i < str.length(); i++) {
            arr[i] = str.charAt(i) - '0';
        }
        return arr;
    }

    // Reverse the digits array
    public static int[] reverseArray(int[] arr) {
        int[] rev = new int[arr.length];
        for (int i = 0; i < arr.length; i++) {
            rev[i] = arr[arr.length - 1 - i];
        }
        return rev;
    }

    // Check if arrays are equal (for palindrome)
    public static boolean isPalindrome(int[] a, int[] b) {
        for (int i = 0; i < a.length; i++) {
            if (a[i] != b[i]) return false;
        }
        return true;
    }

    // Check for duck number
    public static boolean isDuckNumber(int[] digits) {
        for (int i = 1; i < digits.length; i++) {
            if (digits[i] == 0) return true;
        }
        return false;
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java NumberChecker3
Enter a number: 143

--- Results ---
Count of Digits: 3
Is Palindrome: false
Is Duck Number: false
```

5. **Extend or Create a *NumberChecker* utility class and perform following task. Call from main() method the different methods and display results. Make sure all are static methods.**

```java
import java.util.Arrays;
import java.util.Scanner;
public class NumberChecker4 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = input.nextInt();

        System.out.println("\n--- Results ---");
        System.out.println("Is Prime: " + isPrime(number));
        System.out.println("Is Neon Number: " + isNeon(number));
        System.out.println("Is Spy Number: " + isSpy(number));
        System.out.println("Is Automorphic Number: " + isAutomorphic(number));
        System.out.println("Is Buzz Number: " + isBuzz(number));
    }

    // Check if the number is prime
    public static boolean isPrime(int num) {
        if (num <= 1) return false;
        for (int i = 2; i <= num / 2; i++) {
            if (num % i == 0) return false;
        }
        return true;
    }

    // Check if the number is neon
    public static boolean isNeon(int num) {
        int square = num * num;
        int sum = 0;
        while (square > 0) {
            sum += square % 10;
            square /= 10;
        }
        return sum == num;
    }

    // Check if the number is spy
    public static boolean isSpy(int num) {
        int sum = 0, product = 1;
        while (num > 0) {
            int digit = num % 10;
            sum += digit;
            product *= digit;
            num /= 10;
        }
        return sum == product;
    }

    // Check if the number is automorphic
    public static boolean isAutomorphic(int num) {
        int square = num * num;
        return square % (int) Math.pow(10, (int) Math.log10(num) + 1) == num;
    }

    // Check if the number is buzz
    public static boolean isBuzz(int num) {
        return num % 7 == 0 || num % 10 == 7;
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java NumberChecker4
Enter a number: 147

--- Results ---
Is Prime: false
Is Neon Number: false
Is Spy Number: false
Is Automorphic Number: false
Is Buzz Number: true
```

6. **Extend or Create a `NumberChecker` utility class and perform following task. Call from main() method the different methods and display results. Make sure all are static methods.**

```java
import java.util.Arrays;
import java.util.Scanner;
public class NumberChecker5 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = input.nextInt();

        int[] factors = getFactors(number);

        System.out.println("\n--- Results ---");
        System.out.println("Factors: ");
        for (int factor : factors) {
            System.out.print(factor + " ");
        }
        System.out.println("\nGreatest Factor: " + greatestFactor(factors));
        System.out.println("Sum of Factors: " + sumOfFactors(factors));
        System.out.println("Product of Factors: " + productOfFactors(factors));
        System.out.println("Product of Cubes of Factors: " + productOfCubes(factors));
        System.out.println("Is Perfect Number: " + isPerfectNumber(number));
        System.out.println("Is Abundant Number: " + isAbundantNumber(number));
        System.out.println("Is Deficient Number: " + isDeficientNumber(number));
        System.out.println("Is Strong Number: " + isStrongNumber(number));

    }

    // Get factors of a number
    public static int[] getFactors(int num) {
        int count = 0;
        for (int i = 1; i <= num / 2; i++) {
            if (num % i == 0) count++;
        }

        int[] factors = new int[count];
        int index = 0;
        for (int i = 1; i <= num / 2; i++) {
            if (num % i == 0) factors[index++] = i;
        }
        return factors;
    }
```

```java
// Greatest factor
public static int greatestFactor(int[] factors) {
    int max = 0;
    for (int factor : factors) {
        if (factor > max) max = factor;
    }
    return max;
}

// Sum of factors
public static int sumOfFactors(int[] factors) {
    int sum = 0;
    for (int factor : factors) {
        sum += factor;
    }
    return sum;
}

// Product of factors
public static int productOfFactors(int[] factors) {
    int product = 1;
    for (int factor : factors) {
        product *= factor;
    }
    return product;
}

// Product of cubes of the factors
public static double productOfCubes(int[] factors) {
    double product = 1;
    for (int factor : factors) {
        product *= Math.pow(factor, 3);
    }
    return product;
}
```

```java
    // Check if number is perfect
    public static boolean isPerfectNumber(int num) {
        return sumOfFactors(getFactors(num)) == num;
    }

    // Check if number is abundant
    public static boolean isAbundantNumber(int num) {
        return sumOfFactors(getFactors(num)) > num;
    }

    // Check if number is deficient
    public static boolean isDeficientNumber(int num) {
        return sumOfFactors(getFactors(num)) < num;
    }

    // Check if number is strong
    public static boolean isStrongNumber(int num) {
        int sum = 0;
        int original = num;
        while (num > 0) {
            int digit = num % 10;
            sum += factorial(digit);
            num /= 10;
        }
        return sum == original;
    }

    // Factorial of a number
    public static int factorial(int num) {
        int fact = 1;
        for (int i = 1; i <= num; i++) {
            fact *= i;
        }
        return fact;
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java NumberChecker5
Enter a number: 664

--- Results ---
Factors:
1 2 4 8 83 166 332
Greatest Factor: 332
Sum of Factors: 596
Product of Factors: 292754944
Product of Cubes of Factors: 2.5090696333749307E25
Is Perfect Number: false
Is Abundant Number: false
Is Deficient Number: true
Is Strong Number: false
```

7. **Write a program to generate a six-digit OTP number using Math.random() method. Validate the numbers are unique by generating the OTP number 10 times and ensuring all the 10 OTPs are not the same.**

```java
import java.util.Arrays;
import java.util.Scanner;
public class OTP {

    public static void main(String[] args) {
        int[] otpNumbers = new int[10];

        // Generate 10 OTPs
        for (int i = 0; i < otpNumbers.length; i++) {
            otpNumbers[i] = generateOTP();
        }

        // Display the generated OTPs
        System.out.println("Generated OTPs:");
        for (int otp : otpNumbers) {
            System.out.println(otp);
        }

        // Check if OTPs are unique
        if (areOTPsUnique(otpNumbers)) {
            System.out.println("\nAll OTPs are unique!");
        } else {
            System.out.println("\nSome OTPs are duplicate!");
        }
    }

    // Method to generate a 6-digit OTP
    public static int generateOTP() {
        return (int) (Math.random() * 900000) + 100000; // 6-digit OTP
    }

    // Method to check if OTPs are unique
    public static boolean areOTPsUnique(int[] otpNumbers) {
        for (int i = 0; i < otpNumbers.length; i++) {
            for (int j = i + 1; j < otpNumbers.length; j++) {
                if (otpNumbers[i] == otpNumbers[j]) {
                    return false; // Found a duplicate OTP
                }
            }
        }
        return true; // All OTPs are unique
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java OTP
Generated OTPs:
357487
446615
349255
822285
149728
599986
271120
335434
721312
416766

All OTPs are unique!
```

8. **Create a program to display a calendar for a given month and year. The program should take the month and year as input from the user and display the calendar for that month. E.g. for 07 2005 user input, the program should display the calendar as shown below.**

```java
import java.util.Arrays;
import java.util.Scanner;
import java.util.Calendar;
public class Calender {

    // Array of month names
    static String[] monthNames = {
        "January", "February", "March", "April", "May", "June",
        "July", "August", "September", "October", "November", "December"
    };

    // Array of days in each month (non-leap year)
    static int[] monthDays = {
        31, 28, 31, 30, 31, 30,
        31, 31, 30, 31, 30, 31
    };

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

        // Get month and year from user
        System.out.print("Enter month (1-12): ");
        int month = scanner.nextInt();
        System.out.print("Enter year: ");
        int year = scanner.nextInt();

        displayCalendar(month, year);
        scanner.close();
    }

    // Method to check for leap year
    public static boolean isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
    }

    // Method to get the number of days in a month
    public static int getDaysInMonth(int month, int year) {
        if (month == 2 && isLeapYear(year)) {
            return 29;
        }
        return monthDays[month - 1];
    }
```

```java
    // Method to get the name of the month
    public static String getMonthName(int month) {
        return monthNames[month - 1];
    }

    // Method to calculate the first day of the month using Gregorian formula
    public static int getFirstDayOfMonth(int day, int month, int year) {
        int y0 = year - (14 - month) / 12;
        int x = y0 + y0 / 4 - y0 / 100 + y0 / 400;
        int m0 = month + 12 * ((14 - month) / 12) - 2;
        int d0 = (day + x + (31 * m0) / 12) % 7;
        return d0; // 0 = Sunday, 1 = Monday, ..., 6 = Saturday
    }

    // Method to display the calendar
    public static void displayCalendar(int month, int year) {
        String monthName = getMonthName(month);
        int daysInMonth = getDaysInMonth(month, year);
        int firstDay = getFirstDayOfMonth(1, month, year);

        System.out.println("\n   " + monthName + " " + year);
        System.out.println("Sun Mon Tue Wed Thu Fri Sat");

        // Print leading spaces for first week
        for (int i = 0; i < firstDay; i++) {
            System.out.print("    ");
        }

        // Print the days of the month
        for (int day = 1; day <= daysInMonth; day++) {
            System.out.printf("%3d ", day);

            // New line after Saturday
            if ((firstDay + day) % 7 == 0) {
                System.out.println();
            }
        }
        System.out.println(); // Move to next line after printing all days
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java Calender
Enter month (1-12): 9
Enter year: 2006

   September 2006
Sun Mon Tue Wed Thu Fri Sat
                    1   2
  3   4   5   6   7   8   9
 10  11  12  13  14  15  16
 17  18  19  20  21  22  23
 24  25  26  27  28  29  30
```

9. **Write a program Euclidean distance between two points as well as the equation of the line using those two points. Use Math functions *Math.pow()* and *Math.sqrt()***

```java
import java.util.Scanner;
public class LineAndDistance {

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

        // Input for two points
        System.out.print("Enter x1: ");
        double x1 = input.nextDouble();
        System.out.print("Enter y1: ");
        double y1 = input.nextDouble();

        System.out.print("Enter x2: ");
        double x2 = input.nextDouble();
        System.out.print("Enter y2: ");
        double y2 = input.nextDouble();

        // Compute and print distance
        double distance = calculateDistance(x1, y1, x2, y2);
        System.out.printf("Euclidean Distance: %.2f\n", distance);

        // Compute and print line equation
        double[] line = findLineEquation(x1, y1, x2, y2);
        System.out.printf("Equation of Line: y = %.2fx + %.2f\n", line[0], line[1]);
    }

    // Method to calculate Euclidean distance
    public static double calculateDistance(double x1, double y1, double x2, double y2) {
        return Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 - y1, 2));
    }

    // Method to calculate slope (m) and intercept (b), return as array
    public static double[] findLineEquation(double x1, double y1, double x2, double y2) {
        double m = (y2 - y1) / (x2 - x1); // slope
        double b = y1 - m * x1;           // y-intercept
        return new double[]{m, b};
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java LineAndDistance
Enter x1: 5
Enter y1: 9
Enter x2: 6
Enter y2: 7
Euclidean Distance: 2.24
Equation of Line: y = -2.00x + 19.00
```

10. Write a program to find the 3 points that are collinear using the slope formulae and area of triangle formulae. check A (2, 4), B (4, 6) and C (6, 8) are Collinear for sampling.

```java
import java.util.Scanner;
public class Triangle {

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

        // Input 3 points
        System.out.print("Enter x1 y1: ");
        int x1 = input.nextInt(), y1 = input.nextInt();

        System.out.print("Enter x2 y2: ");
        int x2 = input.nextInt(), y2 = input.nextInt();

        System.out.print("Enter x3 y3: ");
        int x3 = input.nextInt(), y3 = input.nextInt();

        // Check collinearity
        if (checkBySlope(x1, y1, x2, y2, x3, y3))
            System.out.println("Collinear (Slope Method)");
        else
            System.out.println("Not Collinear (Slope Method)");

        if (checkByArea(x1, y1, x2, y2, x3, y3))
            System.out.println("Collinear (Area Method)");
        else
            System.out.println("Not Collinear (Area Method)");
    }

    static boolean checkBySlope(int x1, int y1, int x2, int y2, int x3, int y3) {
        return (y2 - y1) * (x3 - x2) == (y3 - y2) * (x2 - x1);
    }

    static boolean checkByArea(int x1, int y1, int x2, int y2, int x3, int y3) {
        int area = x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2);
        return area == 0;
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java Triangle
Enter x1 y1: 2 3
Enter x2 y2: 3 6
Enter x3 y3: 6 8
Not Collinear (Slope Method)
Not Collinear (Area Method)
```

11. **Create a program to find the bonus of 10 employees based on their years of service as well as the total bonus amount the 10-year-old company Zara has to pay as a bonus, along with the old and new salary.**

```java
public class Bonus {

    public static void main(String[] args) {
        double[][] employeeData = generateData(); // [salary, years of service]
        double[][] bonusData = calculateBonus(employeeData); // [old salary, bonus, new salary]

        // Print the table and total calculations
        displayResults(bonusData);
    }

    // Generate 10 random employees with salary and years of service
    public static double[][] generateData() {
        double[][] data = new double[10][2]; // [][0] = salary, [][1] = years of service

        for (int i = 0; i < 10; i++) {
            int salary = (int)(Math.random() * 50000) + 50000; // random 5-digit salary (50000 to 99999)
            int years = (int)(Math.random() * 10) + 1; // 1 to 10 years
            data[i][0] = salary;
            data[i][1] = years;
        }
        return data;
    }

    // Calculate bonus and new salary
    public static double[][] calculateBonus(double[][] data) {
        double[][] result = new double[10][3]; // [][0] = old salary, [][1] = bonus, [][2] = new salary

        for (int i = 0; i < 10; i++) {
            double salary = data[i][0];
            int years = (int)data[i][1];
            double bonus = (years > 5) ? 0.05 * salary : 0.02 * salary;
            double newSalary = salary + bonus;

            result[i][0] = salary;
            result[i][1] = bonus;
            result[i][2] = newSalary;
        }

        return result;
    }

    // Display the result table and totals
    public static void displayResults(double[][] result) {
        double totalOld = 0, totalBonus = 0, totalNew = 0;

        System.out.println("Emp\tOld Salary\tBonus\t\tNew Salary");
        System.out.println("-----------------------------------------------------");
        for (int i = 0; i < 10; i++) {
            System.out.printf("%d\t%.2f\t%.2f\t%.2f\n", (i+1), result[i][0], result[i][1], result[i][2]);
            totalOld += result[i][0];
            totalBonus += result[i][1];
            totalNew += result[i][2];
        }

        System.out.println("-----------------------------------------------------");
        System.out.printf("Total\t%.2f\t%.2f\t%.2f\n", totalOld, totalBonus, totalNew);
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java Bonus
Emp     Old Salary      Bonus           New Salary
-------------------------------------------------------
1       99872.00        1997.44 101869.44
2       51088.00        2554.40 53642.40
3       98041.00        4902.05 102943.05
4       64145.00        1282.90 65427.90
5       84731.00        4236.55 88967.55
6       82793.00        1655.86 84448.86
7       75179.00        1503.58 76682.58
8       99043.00        4952.15 103995.15
9       67237.00        1344.74 68581.74
10      73990.00        3699.50 77689.50
-------------------------------------------------------
Total   796119.00       28129.17        824248.17
```

12. **Create a program to take input marks of students in 3 subjects physics, chemistry, and maths. Compute the total, average, and the percentage score .**

```java
import java.util.Scanner;
import java.util.Random;

public class Grade {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Random rand = new Random();

        System.out.print("Enter number of students: ");
        int n = input.nextInt();

        // Arrays to store data
        int[] physics = new int[n];
        int[] chemistry = new int[n];
        int[] math = new int[n];
        int[] total = new int[n];
        double[] average = new double[n];
        double[] percentage = new double[n];
        char[] grade = new char[n];

        // Generate random marks (50-100)
        for (int i = 0; i < n; i++) {
            physics[i] = rand.nextInt(51) + 50;
            chemistry[i] = rand.nextInt(51) + 50;
            math[i] = rand.nextInt(51) + 50;

            // Calculate results
            total[i] = physics[i] + chemistry[i] + math[i];
            average[i] = Math.round((total[i] / 3.0) * 100) / 100.0;
            percentage[i] = average[i];

            // Determine grade
            if (percentage[i] >= 80) grade[i] = 'A';
            else if (percentage[i] >= 70) grade[i] = 'B';
            else if (percentage[i] >= 60) grade[i] = 'C';
            else if (percentage[i] >= 50) grade[i] = 'D';
            else if (percentage[i] >= 40) grade[i] = 'E';
            else grade[i] = 'R';
        }

        // Display results
        System.out.println("\nStudent Score Card");
        System.out.println("---------------------------------------------------------------");
        System.out.println("ID\tPhy\tChem\tMath\tTotal\tAvg\t%\tGrade");
        System.out.println("---------------------------------------------------------------");

        for (int i = 0; i < n; i++) {
            System.out.printf("%d\t%d\t%d\t%d\t%d\t%.2f\t%.2f\t%c\n",
                i+1, physics[i], chemistry[i], math[i],
                total[i], average[i], percentage[i], grade[i]);
        }
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java Grade
Enter number of students: 8

Student Score Card
-----------------------------------------------------------------
ID      Phy     Chem    Math    Total   Avg     %       Grade
-----------------------------------------------------------------
1       98      64      96      258     86.00   86.00   A
2       61      58      95      214     71.33   71.33   B
3       91      100     57      248     82.67   82.67   A
4       71      50      66      187     62.33   62.33   C
5       53      79      94      226     75.33   75.33   B
6       50      69      51      170     56.67   56.67   D
7       54      63      100     217     72.33   72.33   B
8       69      55      61      185     61.67   61.67   C
```

13. **Write a program to perform matrix manipulation operations like addition, subtraction, multiplication, and transpose. Also finding the determinant and inverse of a matrix. The program should take random matrices as input and display the result of the operations.**

```java
import java.util.Random;
public class Matrix {

    public static void main(String[] args) {
        int[][] A = generateMatrix(2, 2); // You can change to 3x3 if neede
        int[][] B = generateMatrix(2, 2);

        System.out.println("Matrix A:");
        printMatrix(A);

        System.out.println("Matrix B:");
        printMatrix(B);

        System.out.println("Addition:");
        printMatrix(addMatrices(A, B));

        System.out.println("Subtraction:");
        printMatrix(subtractMatrices(A, B));

        System.out.println("Multiplication:");
        printMatrix(multiplyMatrices(A, B));

        System.out.println("Transpose of A:");
        printMatrix(transposeMatrix(A));

        System.out.println("Determinant of A: " + determinant2x2(A));
        System.out.println("Inverse of A:");
        double[][] inverse = inverse2x2(A);
        printMatrix(inverse);
    }

    // Generate a random matrix
    public static int[][] generateMatrix(int rows, int cols) {
        Random rand = new Random();
        int[][] mat = new int[rows][cols];
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                mat[i][j] = rand.nextInt(10); // values from 0-9
        return mat;
    }

    // Addition
    public static int[][] addMatrices(int[][] A, int[][] B) {
        int[][] res = new int[A.length][A[0].length];
        for (int i = 0; i < A.length; i++)
            for (int j = 0; j < A[0].length; j++)
                res[i][j] = A[i][j] + B[i][j];
        return res;
    }
```

```java
    // Subtraction
    public static int[][] subtractMatrices(int[][] A, int[][] B) {
        int[][] res = new int[A.length][A[0].length];
        for (int i = 0; i < A.length; i++)
            for (int j = 0; j < A[0].length; j++)
                res[i][j] = A[i][j] - B[i][j];
        return res;
    }

    // Multiplication
    public static int[][] multiplyMatrices(int[][] A, int[][] B) {
        int[][] res = new int[A.length][B[0].length];
        for (int i = 0; i < A.length; i++)
            for (int j = 0; j < B[0].length; j++)
                for (int k = 0; k < B.length; k++)
                    res[i][j] += A[i][k] * B[k][j];
        return res;
    }

    // Transpose
    public static int[][] transposeMatrix(int[][] A) {
        int[][] res = new int[A[0].length][A.length];
        for (int i = 0; i < A.length; i++)
            for (int j = 0; j < A[0].length; j++)
                res[j][i] = A[i][j];
        return res;
    }

    // Determinant for 2x2
    public static int determinant2x2(int[][] A) {
        return A[0][0] * A[1][1] - A[0][1] * A[1][0];
    }

    // Inverse for 2x2
    public static double[][] inverse2x2(int[][] A) {
        int det = determinant2x2(A);
        if (det == 0) {
            System.out.println("Matrix is not invertible.");
            return new double[][]{{0, 0}, {0, 0}};
        }

        double[][] inv = new double[2][2];
        inv[0][0] = A[1][1] / (double) det;
        inv[0][1] = -A[0][1] / (double) det;
        inv[1][0] = -A[1][0] / (double) det;
        inv[1][1] = A[0][0] / (double) det;

        return inv;
    }
```

```java
    // Print int matrix
    public static void printMatrix(int[][] mat) {
        for (int[] row : mat) {
            for (int val : row)
                System.out.print(val + "\t");
            System.out.println();
        }
    }

    // Print double matrix
    public static void printMatrix(double[][] mat) {
        for (double[] row : mat) {
            for (double val : row)
                System.out.printf("%.2f\t", val);
            System.out.println();
        }
    }
}
```

```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 3>java Matrix
Matrix A:
2       5
0       0
Matrix B:
7       2
2       0
Addition:
9       7
2       0
Subtraction:
-5      3
-2      0
Multiplication:
24      4
0       0
Transpose of A:
2       0
5       0
Determinant of A: 0
Inverse of A:
Matrix is not invertible.
0.00    0.00
0.00    0.00
```