Task 1: Find maximum and minimum

```
import java.util.Scanner;
public class MinMaxFinder {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] numbers = new int[10];
        // Accept input from the user
        System.out.print("Enter 10 integers: ");
        for (int i = 0; i < numbers.length; i++) {
            numbers[i] = scanner.nextInt();
        }
        // Initialize min and max values
        int min = numbers[0];
        int max = numbers[0];
        // Find minimum and maximum values
        for (int number : numbers) {
            if (number < min) {
                min = number;
            }
            if (number > max) {
                max = number;
            }
        }
        scanner.close();
        // Display the results
        System.out.println("Minimum value: " + min);
        System.out.println("Maximum value: " + max);
    }
}
```

- The program starts by creating a Scanner object to read input from the user.

- It declares an integer array `numbers` with a size of 10 to store the user's input.
- The program prompts the user to enter 10 integers and stores each number in the `numbers` array.
- It initializes two variables `min` and `max` to the first element of the array.
- A for-each loop iterates over the array `numbers`.
- - If the current number is smaller than `min`, it updates `min`.
- - If the current number is larger than `max`, it updates `max`.
- The program closes the Scanner object to prevent resource leaks.
- Finally, it displays the minimum and maximum values found in the array.

Task 2: Palindrome

```
import java.util.Scanner;

public class Palindrome {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        boolean isPalindrome = true;

        System.out.println("Enter a string: ");

        String userInput = scanner.nextLine().replaceAll("[^a-zA-Z0-9]", "").toLowerCase();

        System.out.println(userInput);

        String reveString = new StringBuilder(userInput).reverse().toString();

        for (int i = 0 ; i < userInput.length() ; i++){

            if (userInput.charAt(i) != reveString.charAt(i) ){

                isPalindrome = false;

            }

        }

        System.out.println(isPalindrome);

        scanner.close();

    }

}
```

- The program starts by creating a Scanner object to read input from the user.
- It declares a boolean variable `isPalindrome` and initializes it to `true` to assume the input is a palindrome initially.
- The user is prompted to enter a string, and the input is processed by:
    - Removing any special characters and spaces using `replaceAll("[^a-zA-Z0-9]", "")`.
    - Converting the string to lowercase using `toLowerCase()`.
- The cleaned input is then stored in the variable `userInput`.

- The program reverses `userInput` by creating a `StringBuilder`, reversing it, and converting it back to a string.
- A `for` loop iterates through each character of `userInput` and compares it with the corresponding character in the reversed string.
    - If any characters don't match, `isPalindrome` is set to `false`.
- The program prints `true` if the input string is a palindrome, or `false` if it isn't.
- Finally, the Scanner object is closed to prevent resource leaks.

Task 3: Create Student Class

```
import java.util.Scanner;

public class Student {

  // Instance variables

  private String name;

  private int age;

  private double[] testScores = new double[3];

  // Constructor

  public Student(String name, int age, double score1, double score2, double score3) {

    this.name = name;

    this.age = age;

    this.testScores[0] = score1;

    this.testScores[1] = score2;

    this.testScores[2] = score3;

  }

  // Method to calculate the average score

  public double calculateAverage() {

    double total = 0;

    for (double score : testScores) {  // Using instance variable testScores

      total += score;

    }

    return total / testScores.length;

  }
```

```
// Method to determine the grade based on the average
  public char determineGrade(double average) {  // Pass the average as a double
     if (average >= 90) { return 'A';  }
     else if (average >= 80) { return 'B';  }
     else if (average >= 70) { return 'C'; }
     else if (average >= 60) { return 'D'; }
     else { return 'F'; }
  }
  // Method to display student details
  public void displayInfo() {
     double studentAve = calculateAverage();
     char studentGrade = determineGrade(studentAve);  // Determine grade based on average
     System.out.println("Student Name: " + name);
     System.out.println("Age: " + age);
     System.out.println("Average Score: " + studentAve);
     System.out.println("Grade: " + studentGrade);
  }
  // Main method to test the Student class
  public static void main(String[] args) {
     // Creating a Student object
     Scanner scanner = new Scanner(System.in);
     System.out.print("Enter the student name: ");
     String studentName = scanner.nextLine();
     System.out.print("Enter the student aga: ");
     int studentAge = scanner.nextInt();
     System.out.print("1st grade: ");
     int grade1 = scanner.nextInt();
     System.out.print("2nd grade: ");
     int grade2 = scanner.nextInt();
     System.out.print("1rd grade: ");
     int grade3 = scanner.nextInt();
     Student student = new Student(studentName, studentAge, grade1, grade2, grade3);
     scanner.close();
     student.displayInfo();  }
}
```

- The program defines a `Student` class with instance variables for the student's name, age, and an array to hold three test scores.
- The constructor takes the student's name, age, and three test scores as parameters and assigns them to the instance variables.
- The `calculateAverage()` method computes the average of the three test scores by iterating through the `testScores` array and dividing the total by the length of the array.
- The `determineGrade()` method takes the calculated average as a parameter and returns a grade ('A', 'B', 'C', etc.) based on predefined thresholds.
- The `displayInfo()` method outputs the student's name, age, average test score, and the corresponding grade.
- In the `main` method, a `Scanner` object is created to accept input from the user for the student's name, age, and three test scores.
- After gathering the input, the `Student` object is created with the provided values.
- The `displayInfo()` method is called to display the student's information, including the calculated average and grade.
- The `Scanner` object is closed to avoid memory leaks.

Task 4: 3x3 Matrices

```
import java.util.Scanner;

public class MatrixAddition {

   public static void main(String[] args) {

      Scanner scanner = new Scanner(System.in);

      // Declare two 3x3 matrices

      int[][] matrix1 = new int[3][3];

      int[][] matrix2 = new int[3][3];

      int[][] sumMatrix = new int[3][3];

      // Input for the first matrix

      System.out.println("Enter elements of the first 3x3 matrix:");

      for (int i = 0; i < 3; i++) {

         for (int j = 0; j < 3; j++) {

            System.out.print("Element [" + i + "][" + j + "]: ");

            matrix1[i][j] = scanner.nextInt();

         }

      }
```

```
// Input for the second matrix
        System.out.println("Enter elements of the second 3x3 matrix:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print("Element [" + i + "][" + j + "]: ");
                matrix2[i][j] = scanner.nextInt();
            }
        }
        // Perform matrix addition
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }
        // Display the result
        System.out.println("Sum of the two matrices:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(sumMatrix[i][j] + "\t"); // Tab space for better formatting
            }
            System.out.println(); // New line after each row
        }
        scanner.close(); // Close the scanner
    }
}
```

Task 5: Bubble Sort

```
import java.util.Scanner;
public class BubbleSort {
   public static void main(String[] args) {
      Scanner user_input = new Scanner(System.in);
      int[] arrayOfNums = new int[5];
      System.out.println("Enter 5 numbers: ");
      for (int i = 0 ; i < arrayOfNums.length; i++){
         arrayOfNums[i] = user_input.nextInt();
      }
      for (int i = 0 ; i < arrayOfNums.length ; i++) {
         for (int j = 0 ; j < arrayOfNums.length - 1 ; j++ ){
            if (arrayOfNums[i] < arrayOfNums[j]) {
               int temp = arrayOfNums[i];
               arrayOfNums[i] = arrayOfNums[j];
               arrayOfNums[j] = temp;
            }
         }
      }
      System.out.println("Sorted Array: ");
      for (int x : arrayOfNums) {
         System.out.print(x + " ");
      }
      user_input.close();
   }
}
```

- The program starts by creating a Scanner object to read input from the user.
- It declares an array named `arrayOfNums` to hold 5 integers.
- The user is prompted to enter 5 numbers, which are stored in the `arrayOfNums` array.
- A nested for loop is used to perform the bubble sort algorithm.

    - The outer loop iterates through each element of the array.
    - The inner loop compares adjacent elements in the array and swaps them if they are in the wrong order.
- Inside the inner loop, if the current element is smaller than the next element, they are swapped using a temporary variable `temp`.
- Once the sorting is complete, the program prints the sorted array in ascending order.
- Finally, the program closes the Scanner object to prevent resource leaks.