Level 2 Practice Programs

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

   Hint =>

   a. Zara decides to give a bonus of 5% to employees whose year of service is more than 5 years or 2% if less than 5 years
   b. Define a double array to save salary and years of service for each of the 10 employees
   c. Also define a double array to save the new salary and the bonus amount as well as variables to save the total bonus, total old salary, and new salary
   d. Define a loop to take input from the user. If salary or year of service is an invalid number then ask the use to enter again. Note in this case you will have to decrement the index counter
   e. Define another loop to calculate the bonus of 10 employees based on their years of service. Save the bonus in the array, compute the new salary, and save in the array. Also, the total bonus and total old and new salary can be calculated in the loop
   f. Print the total bonus payout as well as the total old and new salary of all the employees

```java
import java.util.Scanner;

public class employeeBonus {

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

       double[] salaries = new double[10];
       int[] yearsOfService = new int[10];
       double[] bonuses = new double[10];
       double[] newSalaries = new double[10];

       double totalBonus = 0;
       double totalOldSalary = 0;
       double totalNewSalary = 0;

       System.out.println("Enter salary and years of service for 10 employees:");

       for (int i = 0; i < 10; i++) {
           boolean validInput = false;
           while (!validInput) {
               System.out.print("Employee " + (i + 1) + " Salary: ");
               if (!input.hasNextDouble()) {
                   System.out.println("Invalid input. Please enter a valid
salary.");
                   input.next(); // Clear invalid input
                   continue;
               }
               salaries[i] = input.nextDouble();

               System.out.print("Employee " + (i + 1) + " Years of Service: ");
               if (!input.hasNextInt()) {
                   System.out.println("Invalid input. Please enter a valid number
of years.");
                   input.next(); // Clear invalid input
                   continue;
               }
               yearsOfService[i] = input.nextInt();

               validInput = true;
           }
       }

       for (int i = 0; i < 10; i++) {
           double bonusPercentage = (yearsOfService[i] > 5) ? 0.05 : 0.02;
           bonuses[i] = salaries[i] * bonusPercentage;
           newSalaries[i] = salaries[i] + bonuses[i];

           totalBonus += bonuses[i];
           totalOldSalary += salaries[i];
           totalNewSalary += newSalaries[i];
       }

       System.out.println("\nTotal Bonus Payout: $" + totalBonus);
       System.out.println("Total Old Salary: $" + totalOldSalary);
       System.out.println("Total New Salary: $" + totalNewSalary);

       input.close();
   }
}
```

2. Create a program to find the youngest friends among 3 Amar, Akbar, and Anthony based on their ages and the tallest among the friends based on their heights

   Hint =>

   a. Take user input for age and height for the 3 friends and store it in two arrays each to store the values for age and height of the 3 friends
   b. Loop through the array and find the youngest of the 3 friends and the tallest of the 3 friends
   c. Finally display the youngest and tallest of the 3 friends

```
import java.util.Scanner;

public class youngestTallest {

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

        String[] names = {"Amar", "Akbar", "Anthony"};
        int[] ages = new int[3];
        double[] heights = new double[3];

        // Input ages and heights
        for (int i = 0; i < 3; i++) {
            System.out.print("Enter age for " + names[i] + ": ");
            if (!input.hasNextInt()) {
                System.out.println("Invalid input. Please enter an integer for
age.");
                input.next();
                i--;
                continue;
            }
            ages[i] = input.nextInt();

            System.out.print("Enter height for " + names[i] + " (in meters): ");
            if (!input.hasNextDouble()) {
                System.out.println("Invalid input. Please enter a double for
height.");
                input.next();
                i--;
                continue;
            }
            heights[i] = input.nextDouble();
        }

        // Find youngest
        int youngestIndex = 0;
        for (int i = 1; i < 3; i++) {
            if (ages[i] < ages[youngestIndex]) {
                youngestIndex = i;
            }
        }

        // Find tallest
        int tallestIndex = 0;
        for (int i = 1; i < 3; i++) {
            if (heights[i] > heights[tallestIndex]) {
                tallestIndex = i;
            }
        }

        // Display results
        System.out.println("\nYoungest friend: " + names[youngestIndex] + " (Age: " + ages[youngestIndex] + ")");
        System.out.println("Tallest friend: " + names[tallestIndex] + " (Height: " + heights[tallestIndex] + " meters)");

        input.close();
    }
}
```

3. Create a program to store the digits of the number in an array and find the largest and second largest element of the array.

   Hint =>

   a. Create a number variable and Take user input.
   b. Define an array to store the digits. Set the size of the array to maxDigit variable initially set to 10
   c. Create an integer variable index with the value 0 to reflect the array index.
   d. Use a loop to iterate until the number is not equal to 0.
   e. Remove the last digit from the number in each iteration and add it to the array.
   f. Increment the index by 1 in each iteration and if the index count equals maxDigit then break out of the loop and the remaining digits are not added to the array
   g. Define variable to store largest and second largest digit and initialize it to zero
   h. Loop through the array and use conditional statements to find the largest and second largest number in the array
   i. Finally display the largest and second-largest number

```
import java.util.Scanner;

public class largestSecondLargestDigits {

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

        System.out.print("Enter a number: ");
```

```
if (!input.hasNextInt()) {
    System.out.println("Invalid input. Please enter an integer.");
    input.close();
    return;
}

int number = input.nextInt();

if (number < 0) {
    System.out.println("Please enter a non-negative integer.");
    input.close();
    return;
}

int maxDigit = 10;
int[] digits = new int[maxDigit];
int index = 0;

while (number != 0) {
    digits[index] = number % 10;
    number /= 10;
    index++;

    if (index == maxDigit) {
        break;
    }
}

int largest = Integer.MIN_VALUE;
int secondLargest = Integer.MIN_VALUE;
```

```
        for (int i = 0; i < index; i++) {
            if (digits[i] > largest) {
                secondLargest = largest;
                largest = digits[i];
            } else if (digits[i] > secondLargest && digits[i] != largest) {
                secondLargest = digits[i];
            }
        }

        if (index == 0) {
            System.out.println("No digits to analyze.");
        } else if (index == 1) {
            System.out.println("Largest digit: " + largest);
            System.out.println("Second largest digit: Not found.");
        } else {
            System.out.println("Largest digit: " + largest);
            if (secondLargest == Integer.MIN_VALUE) {
                System.out.println("Second largest digit: Not found.");
            } else {
                System.out.println("Second largest digit: " + secondLargest);
            }
        }

        input.close();
    }
}
```

4. Rework the program 2, especially the **Hint** f where if index equals maxDigit, we break from the loop. Here we want to modify to Increase the size of the array i,e maxDigit by 10 if the index is equal to maxDigit. This is done to consider all digits to find the largest and second-largest number

Hint =>

a. In Hint f inside the loop if the index is equal to maxDigit, increase maxDigit and make digits array to store more elements.

b. To do this, we need to create a new temp array of size maxDigit, copy from the current digits array the digits into the temp array, and assign the current digits array to the temp array

c. Now the digits array will be able to store all digits of the number in the array and then find the largest and second largest number

```java
import java.util.Scanner;

public class largestSecondLargestDigitsDynamic {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number: ");

        if (!input.hasNextInt()) {

            System.out.println("Invalid input. Please enter an integer.");

            input.close();

            return;

        }

        int number = input.nextInt();

        if (number < 0) {

            System.out.println("Please enter a non-negative integer.");

            input.close();

            return;

        }

        int maxDigit = 10;

        int[] digits = new int[maxDigit];

        int index = 0;

        while (number != 0) {

            digits[index] = number % 10;

            number /= 10;

            index++;

            if (index == maxDigit) {

                maxDigit += 10; // Increase maxDigit by 10

                int[] temp = new int[maxDigit];

                System.arraycopy(digits, 0, temp, 0, digits.length);
                digits = temp; // Update digits array
            }
        }
        int largest = Integer.MIN_VALUE;
        int secondLargest = Integer.MIN_VALUE;
        for (int i = 0; i < index; i++) {
            if (digits[i] > largest) {
                secondLargest = largest;
                largest = digits[i];
            } else if (digits[i] > secondLargest && digits[i] != largest) {
                secondLargest = digits[i];
            }
        }
        if (index == 0) {
            System.out.println("No digits to analyze.");
        } else if (index == 1) {
            System.out.println("Largest digit: " + largest);
            System.out.println("Second largest digit: Not found.");
        } else {
            System.out.println("Largest digit: " + largest);
            if (secondLargest == Integer.MIN_VALUE) {
                System.out.println("Second largest digit: Not found.");
            } else {
                System.out.println("Second largest digit: " + secondLargest);
            }
        }
        input.close();
    }
}
```

5. Create a program to take a number as input and reverse the number. To do this, store the digits of the number in an array and display the array in reverse order

Hint =>

a. Take user input for a number.
b. Find the count of digits in the number.
c. Find the digits in the number and save them in an array
d. Create an array to store the elements of the digits array in reverse order
e. Finally, display the elements of the array in reverse order

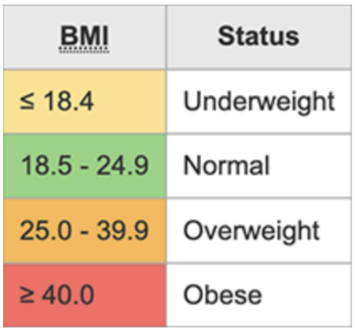
```java
import java.util.Scanner;

public class reverseNumber {

   public static void main(String[] args) {

       Scanner input = new Scanner(System.in);

       System.out.print("Enter a non-negative integer: ");

       if (!input.hasNextInt()) {

           System.out.println("Invalid input. Please enter an integer.");

           input.close();

           return;

       }

       int number = input.nextInt();

       if (number < 0) {

           System.out.println("Please enter a non-negative integer.");

           input.close();

           return;

       }

       int tempNumber = number;

       int digitCount = 0;

       // Count digits

       if (tempNumber == 0) {

           digitCount = 1;

       } else {

           while (tempNumber != 0) {

               tempNumber /= 10;

               digitCount++;

           }

       }
```

```
        int[] digits = new int[digitCount];
        int[] reversedDigits = new int[digitCount];
        tempNumber = number; // Reset tempNumber
        int index = 0;
        // Extract digits
        while (tempNumber != 0) {
            digits[index] = tempNumber % 10;
            tempNumber /= 10;
            index++;
        }
        // Reverse the array
        for (int i = 0; i < digitCount; i++) {
            reversedDigits[i] = digits[digitCount - 1 - i];
        }
        // Display reversed digits
        System.out.print("Reversed number: ");
        for (int digit : reversedDigits) {
            System.out.print(digit);
        }
        System.out.println();
        input.close();
    }
}
```

6. An organization took up an exercise to find the Body Mass Index (BMI) of all the persons in the team. For this create a program to find the BMI and display the height, weight, BMI and status of each individual

   Hint =>

   a. Take input for a number of persons
   b. Create arrays to store the weight, height, BMI, and weight status of the persons
   c. Take input for the weight and height of the persons
   d. Calculate the BMI of all the persons and store them in an array and also find the weight status of the persons
   e. Display the height, weight, BMI, and weight status of each person
   f. Use the table to determine the weight status of the person



BMI	Status
≤ 18.4	Underweight
18.5 - 24.9	Normal
25.0 - 39.9	Overweight
≥ 40.0	Obese

```java
import java.util.Scanner;
public class BMIAnalyzer {
   public static void main(String[] args) {
       Scanner input = new Scanner(System.in);
       System.out.print("Enter the number of persons: ");
       if (!input.hasNextInt()) {
           System.out.println("Invalid input. Please enter an integer.");
           input.close();
           return;
       }
       int numPersons = input.nextInt();
       if (numPersons <= 0) {
           System.out.println("Please enter a positive integer for the number of persons.");
           input.close();
           return;
       }
       double[] weights = new double[numPersons];
       double[] heights = new double[numPersons];
       double[] bmis = new double[numPersons];
       String[] statuses = new String[numPersons];
       // Input weight and height
       for (int i = 0; i < numPersons; i++) {
           System.out.print("Enter weight (kg) for person " + (i + 1) + ": ");
```

```
            if (!input.hasNextDouble()) {
                System.out.println("Invalid input. Please enter a valid weight
(kg).");
                input.next();
                i--;
                continue;
            }
            weights[i] = input.nextDouble();
            System.out.print("Enter height (m) for person " + (i + 1) + ": ");
            if (!input.hasNextDouble()) {
                System.out.println("Invalid input. Please enter a valid height
(m).");
                input.next();
                i--;
                continue;
            }
            heights[i] = input.nextDouble();
        }
        // Calculate BMI and status
        for (int i = 0; i < numPersons; i++) {
            bmis[i] = weights[i] / (heights[i] * heights[i]);
            statuses[i] = getBMIStatus(bmis[i]);
        }
        // Display results
        System.out.println("\nBMI Analysis:");
System.out.println("------------------------------------------------------");
        System.out.printf("%-10s %-10s %-10s %-20s\n", "Height (m)", "Weight
(kg)", "BMI", "Weight Status");
System.out.println("------------------------------------------------------");
        for (int i = 0; i < numPersons; i++) {
            System.out.printf("%-10.2f %-10.2f %-10.2f %-20s\n", heights[i],
weights[i], bmis[i], statuses[i]);
        }
        input.close();
    }
    public static String getBMIStatus(double bmi) {
        if (bmi < 18.5) {
            return "Underweight";
        } else if (bmi < 25) {
            return "Normal weight";
        } else if (bmi < 30) {
            return "Overweight";
        } else {
            return "Obese";
        }
    }
}
```

7. Rewrite the above program using multi-dimensional array to store height, weight, and BMI in 2D array for all the persons

   Hint =>

   a. Take input for a number of persons
   b. Create a multi-dimensional array to store weight, height and BMI. Also create an to store the weight status of the persons

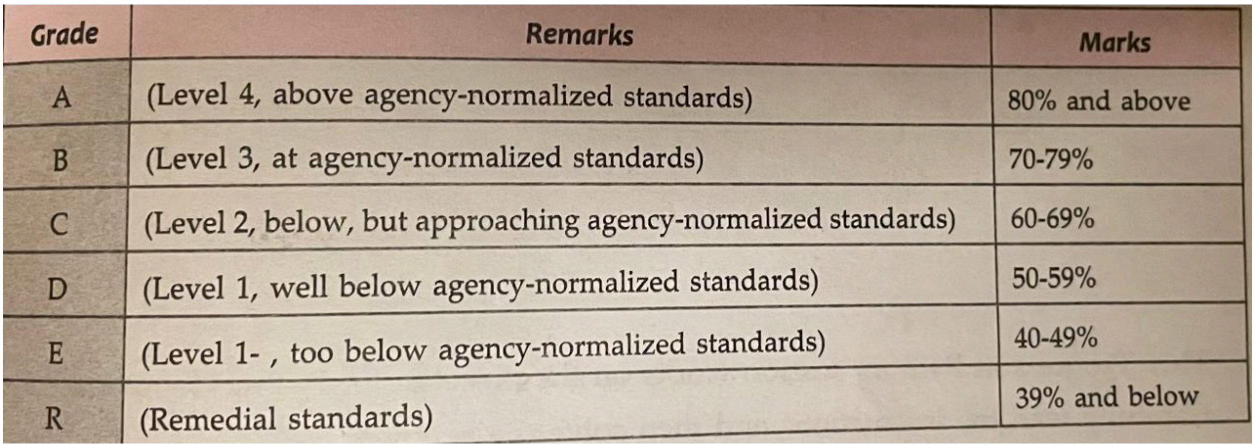
   ```
   double[][] personData = new double[number][3];
   String[] weightStatus = new String[number];
   ```

   c. Take input for weight and height of the persons and for negative values, ask the user to enter positive values
   d. Calculate BMI of all the persons and store them in the personData array and also find the weight status and put them in the weightStatus array
   e. Display the height, weight, BMI and status of each person

```
import java.util.Scanner;
public class BMIAnalyzer2D {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the number of persons: ");
        if (!input.hasNextInt()) {
            System.out.println("Invalid input. Please enter an integer.");
```

```
            input.close();
            return;
        }
        int numPersons = input.nextInt();
        if (numPersons <= 0) {
            System.out.println("Please enter a positive integer for the number of
persons.");
            input.close();
            return;
        }
        double[][] personData = new double[numPersons][3]; // [persons][weight,
height, BMI]
        String[] weightStatus = new String[numPersons];
        // Input weight and height
        for (int i = 0; i < numPersons; i++) {
            boolean validInput = false;
            while (!validInput) {
                System.out.print("Enter weight (kg) for person " + (i + 1) + ": ");
                if (!input.hasNextDouble()) {
                    System.out.println("Invalid input. Please enter a valid weight
(kg).");
                    input.next();
                    continue;
                }
                double weight = input.nextDouble();
                if (weight <= 0) {
                    System.out.println("Weight must be positive. Please re-enter.");
                    continue;
                }
                personData[i][0] = weight;

                System.out.print("Enter height (m) for person " + (i + 1) + ": ");
                if (!input.hasNextDouble()) {
                    System.out.println("Invalid input. Please enter a valid height
(m).");
                    input.next();
                    continue;
                }
                double height = input.nextDouble();
                if (height <= 0) {
                    System.out.println("Height must be positive. Please re-enter.");
                    continue;
                }
                personData[i][1] = height;
                validInput = true;
            }
        }
        // Calculate BMI and status
        for (int i = 0; i < numPersons; i++) {
            personData[i][2] = personData[i][0] / (personData[i][1] *
personData[i][1]);
            weightStatus[i] = getBMIStatus(personData[i][2]);
        }
        // Display results
        System.out.println("\nBMI Analysis:");
System.out.println("-----------------------------------------------------");
        System.out.printf("%-10s %-10s %-10s %-20s\n", "Height (m)", "Weight (kg)",
"BMI", "Weight Status");
System.out.println("-----------------------------------------------------");
        for (int i = 0; i < numPersons; i++) {
            System.out.printf("%-10.2f %-10.2f %-10.2f %-20s\n", personData[i][1],
personData[i][0], personData[i][2], weightStatus[i]);
        }
        input.close();
    }
    public static String getBMIStatus(double bmi) {
        if (bmi < 18.5) {
            return "Underweight";
        } else if (bmi < 25) {
            return "Normal weight";
        } else if (bmi < 30) {
            return "Overweight";
        } else {
            return "Obese";
        }
    }
}
```

8. Create a program to take input marks of students in 3 subjects physics, chemistry, and maths. Compute the percentage and then calculate the grade as per the following guidelines



Grade	Remarks	Marks
A	(Level 4, above agency-normalized standards)	80% and above
B	(Level 3, at agency-normalized standards)	70-79%
C	(Level 2, below, but approaching agency-normalized standards)	60-69%
D	(Level 1, well below agency-normalized standards)	50-59%
E	(Level 1- , too below agency-normalized standards)	40-49%
R	(Remedial standards)	39% and below

Hint =>

a. Take input for the number of students
b. Create arrays to store marks, percentages, and grades of the students
c. Take input for marks of students in physics, chemistry, and maths. If the marks are negative, ask the user to enter positive values and decrement the index
d. Calculate the percentage and grade of the students based on the percentage
e. Display the marks, percentages, and grades of each student

```
import java.util.Scanner;


public class gradeCalculator {

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

       System.out.print("Enter the number of students: ");

       if (!input.hasNextInt()) {
           System.out.println("Invalid input. Please enter an integer.");
           input.close();
           return;
       }

       int numStudents = input.nextInt();

       if (numStudents <= 0) {
           System.out.println("Please enter a positive integer for the number of students.");
           input.close();
           return;
       }

       int[][] marks = new int[numStudents][3]; // [students][physics, chemistry, maths]
       double[] percentages = new double[numStudents];
       String[] grades = new String[numStudents];

       // Input marks for each student
       for (int i = 0; i < numStudents; i++) {
           boolean validInput = false;
           while (!validInput) {
               System.out.print("Enter Physics marks for student " + (i + 1) + ": ");
               if (!input.hasNextInt()) {
```

```
                System.out.println("Invalid input. Please enter an integer.");
                input.next();
                continue;
            }
            int physics = input.nextInt();
            if (physics < 0) {
                System.out.println("Marks must be non-negative. Please
re-enter.");
                continue;
            }
            marks[i][0] = physics;

            System.out.print("Enter Chemistry marks for student " + (i + 1) + ":
");
            if (!input.hasNextInt()) {
                System.out.println("Invalid input. Please enter an integer.");
                input.next();
                continue;
            }
            int chemistry = input.nextInt();
            if (chemistry < 0) {
                System.out.println("Marks must be non-negative. Please
re-enter.");
                continue;
            }
            marks[i][1] = chemistry;

            System.out.print("Enter Maths marks for student " + (i + 1) + ": ");
            if (!input.hasNextInt()) {
                System.out.println("Invalid input. Please enter an integer.");
                input.next();
                continue;
                }
                int maths = input.nextInt();
                if (maths < 0) {
                    System.out.println("Marks must be non-negative. Please
re-enter.");
                    continue;
                }
                marks[i][2] = maths;

                validInput = true;
            }
        }

        // Calculate percentage and grade
        for (int i = 0; i < numStudents; i++) {
            int totalMarks = marks[i][0] + marks[i][1] + marks[i][2];
            percentages[i] = (double) totalMarks / 3;
            grades[i] = calculateGrade(percentages[i]);
        }

        // Display results
        System.out.println("\nStudent Grades:");

System.out.println("---------------------------------------------------------------
-------------------");
        System.out.printf("%-10s %-10s %-10s %-15s %-10s %-40s\n", "Student",
"Physics", "Chemistry", "Maths", "Percentage", "Grade and Remarks");

System.out.println("---------------------------------------------------------------
-------------------");


        for (int i = 0; i < numStudents; i++) {
            System.out.printf("%-10d %-10d %-10d %-15d %-10.2f %-40s\n", (i + 1), marks[i][0], marks[i][1], marks[i][2], percentages[i], grades[i] + getRemarks(grades[i]));
        }

        input.close();
    }

    public static String calculateGrade(double percentage) {
        if (percentage >= 80) {
            return "A";
        } else if (percentage >= 70) {
            return "B";
        } else if (percentage >= 60) {
            return "C";
        } else if (percentage >= 50) {
            return "D";
        } else if (percentage >= 40) {
            return "E";
        } else {
            return "R";
        }
    }

    public static String getRemarks(String grade) {
        switch (grade) {
            case "A":
                return " (Level 4, above agency-normalized standards)";
            case "B":
                return " (Level 3, at agency-normalized standards)";
            case "C":
                return " (Level 2, below, but approaching agency-normalized
standards)";
            case "D":
                return " (Level 1, well below agency-normalized standards)";
            case "E":
                return " (Level 1-, too below agency-normalized standards)";
            case "R":
                return " (Remedial standards)";
            default:
                return "";
        }
    }
}
```

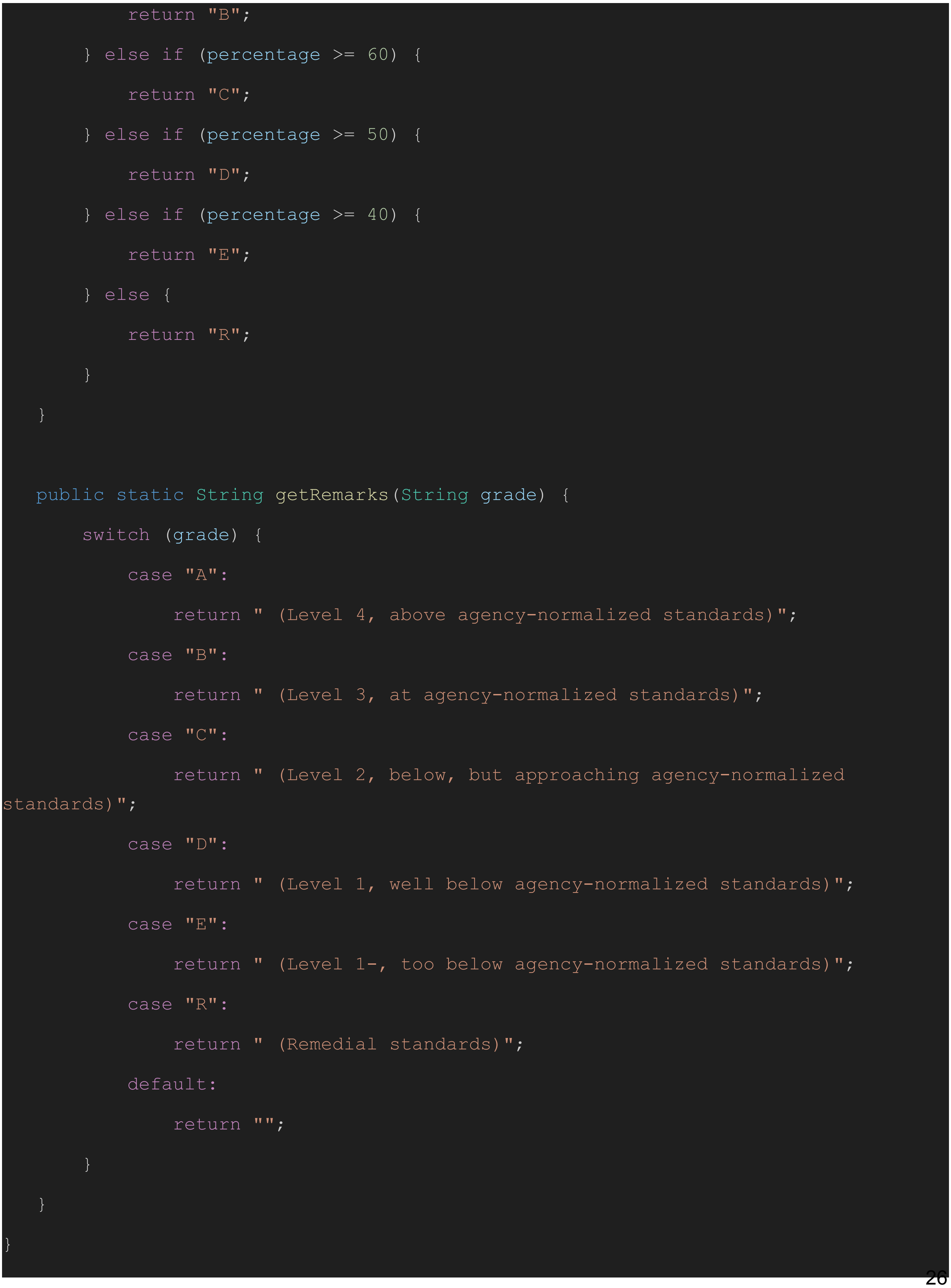
9. Rewrite the above program to store the marks of the students in physics, chemistry, and maths in a 2D array and then compute the percentage and grade

    Hint =>

    a. All the steps are the same as the problem 8 except the marks are stored in a 2D array
    b. Use the 2D array to calculate the percentages, and grades of the students

```
import java.util.Scanner;

public class gradeCalculator2D {

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

        System.out.print("Enter the number of students: ");

        if (!input.hasNextInt()) {
            System.out.println("Invalid input. Please enter an integer.");
            input.close();
            return;
        }
        int numStudents = input.nextInt();

        if (numStudents <= 0) {
            System.out.println("Please enter a positive integer for the number of students.");
            input.close();
            return;
        }

        int[][] marks = new int[numStudents][3]; // [students][physics, chemistry, maths]
        double[] percentages = new double[numStudents];
        String[] grades = new String[numStudents];

        // Input marks for each student
        for (int i = 0; i < numStudents; i++) {
            boolean validInput = false;
            while (!validInput) {
                System.out.print("Enter Physics marks for student " + (i + 1) + ": ");
                if (!input.hasNextInt()) {
                    System.out.println("Invalid input. Please enter an integer.");
                    input.next();
                    continue;
                }
                int physics = input.nextInt();
                if (physics < 0) {
                    System.out.println("Marks must be non-negative. Please re-enter.");
                    continue;
                }
                marks[i][0] = physics;
                System.out.print("Enter Chemistry marks for student " + (i + 1) + ":
");
                if (!input.hasNextInt()) {
                    System.out.println("Invalid input. Please enter an integer.");
                    input.next();
                    continue;
                }
                int chemistry = input.nextInt();
                if (chemistry < 0) {
                    System.out.println("Marks must be non-negative. Please
re-enter.");
                    continue;
                }
                marks[i][1] = chemistry;

                System.out.print("Enter Maths marks for student " + (i + 1) + ": ");
                if (!input.hasNextInt()) {
                    System.out.println("Invalid input. Please enter an integer.");
                    input.next();
                    continue;
                }
                int maths = input.nextInt();
                if (maths < 0) {
                    System.out.println("Marks must be non-negative. Please
re-enter.");
                    continue;
                }
                marks[i][2] = maths;

                validInput = true;
            }
```

```
        }

        // Calculate percentage and grade
        for (int i = 0; i < numStudents; i++) {
            int totalMarks = marks[i][0] + marks[i][1] + marks[i][2];
            percentages[i] = (double) totalMarks / 3;
            grades[i] = calculateGrade(percentages[i]);
        }

        // Display results
        System.out.println("\nStudent Grades:");
        System.out.println("----------------------------------------------------------------------------------");
        System.out.printf("%-10s %-10s %-10s %-15s %-10s %-40s\n", "Student", "Physics", "Chemistry", "Maths", "Percentage", "Grade and Remarks");
        System.out.println("----------------------------------------------------------------------------------");

        for (int i = 0; i < numStudents; i++) {
            System.out.printf("%-10d %-10d %-10d %-15d %-10.2f %-40s\n", (i + 1), marks[i][0], marks[i][1], marks[i][2], percentages[i], grades[i] + getRemarks(grades[i]));
        }

        input.close();
    }

    public static String calculateGrade(double percentage) {
        if (percentage >= 80) {
            return "A";
        } else if (percentage >= 70) {
```



```
            return "B";
        } else if (percentage >= 60) {
            return "C";
        } else if (percentage >= 50) {
            return "D";
        } else if (percentage >= 40) {
            return "E";
        } else {
            return "R";
        }
    }

    public static String getRemarks(String grade) {
        switch (grade) {
            case "A":
                return " (Level 4, above agency-normalized standards)";
            case "B":
                return " (Level 3, at agency-normalized standards)";
            case "C":
                return " (Level 2, below, but approaching agency-normalized
standards)";
            case "D":
                return " (Level 1, well below agency-normalized standards)";
            case "E":
                return " (Level 1-, too below agency-normalized standards)";
            case "R":
                return " (Remedial standards)";
            default:
                return "";
        }
    }
}
```

10. Create a program to take a number as input find the frequency of each digit in the number using an array and display the frequency of each digit

Hint =>

a. Take the input for a number
b. Find the count of digits in the number
c. Find the digits in the number and save them in an array
d. Find the frequency of each digit in the number. For this define a frequency array of size 10, Loop through the digits array, and increase the frequency of each digit
e. Display the frequency of each digit in the number

```java
import java.util.Scanner;

public class digitFrequency {

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

        System.out.print("Enter a non-negative integer: ");

        if (!input.hasNextInt()) {
            System.out.println("Invalid input. Please enter an integer.");
            input.close();
            return;
        }

        int number = input.nextInt();

        if (number < 0) {
            System.out.println("Please enter a non-negative integer.");
            input.close();
            return;
        }

        int tempNumber = number;
        int digitCount = 0;

        // Count digits
        if (tempNumber == 0) {
            digitCount = 1;
        } else {
            while (tempNumber != 0) {
                tempNumber /= 10;
                digitCount++;
            }
        }

        int[] digits = new int[digitCount];
        int index = 0;
        tempNumber = number; // Reset tempNumber

        // Extract digits
        while (tempNumber != 0) {
            digits[index] = tempNumber % 10;
            tempNumber /= 10;
            index++;
        }

        int[] frequency = new int[10]; // Frequency array for digits 0-9

        // Calculate frequency
        for (int digit : digits) {
            frequency[digit]++;
        }
```

```java
        // Display frequency
        System.out.println("Digit frequencies:");
        for (int i = 0; i < 10; i++) {
            if (frequency[i] > 0) {
                System.out.println("Digit " + i + ": " + frequency[i]);
            }
        }

        input.close();
    }
}
```