Part 1: Data Types, Variables, and Arithmetic Operations
1. Declare variables of different data types (int, double, boolean, char) and perform various arithmetic operations on them.
2. Write a program that calculates the area and perimeter of a rectangle, given the length and width as input.
3. Implement a program that converts temperature from Celsius to Fahrenheit and vice versa.

```
class ddt {
    public static void main(String[] args) {
        // Declare variables
        int a = 10;
        int b = 20;
        double c = 15.5;
        boolean t = true;
        char ch = 'A';

        // Perform arithmetic operations
        int s = add(a, b);
        double p = mul(a, c);
        double d = sub(c, a);
        double div = div(c, b);

        // Output results
        System.out.println("Sum of a and b: " + s);
        System.out.println("Product of a and c: " + p);
        System.out.println("Difference of c and a: " + d);
        System.out.println("Division of c by b: " + div);
        System.out.println("Boolean value: " + t);
        System.out.println("Character: " + ch);
    }

    // Method to add two integers
    public static int add(int x, int y) {
        return x + y;
    }

    // Method to multiply an int and a double
    public static double mul(int x, double y) {
        return x * y;
    }

    // Method to subtract an int from a double
    public static double sub(double x, int y) {
        return x - y;
    }

    // Method to divide a double by an int
    public static double div(double x, int y) {
        return x / y;
    }
}
```

```
Output:

Sum of a and b: 30
Product of a and c: 155.0
Difference of c and a: 5.5
Division of c by b: 0.775
Boolean value: true
Character: A
```

—---

2)

```
import java.util.Scanner;

class hello{
      public static void main(String args[]){
            Scanner s1 = new Scanner(System.in);

            System.out.print("Enter length : ");
            int len = s1.nextInt();

            System.out.print("Enter width : ");
            int wid = s1.nextInt();

            System.out.println("Area of Rectangle : " + len*wid);
            System.out.println("Perimeter of Rectangle : " + (2*(len+
wid)));
      }
}

Output:
```

```
Enter length : 5
Enter width : 3

Area of Rectangle : 15
Perimeter of Rectangle : 16
```

—---

3)

```
import java.util.Scanner;

class hello{
      public static double celsius_to_fahrenheit(double cel){
            return (cel* 9/5 )+32;
      }
```

```
	public static double fahrenheit_to_celsius(double far){
		return (far-32)*5/9;
	}

	public static void main(String args[]){
		Scanner s1 = new Scanner(System.in);

		System.out.print("Enter Celsius : ");
		double cel = s1.nextDouble();

		System.out.println("celsius : " + cel);
		System.out.println("Celsius to Fahrenheit : " +
celsius_to_fahrenheit(cel));

		System.out.print("Enter Fahrenheit : ");
		double far = s1.nextDouble();

		System.out.println("celsius : " + far);
		System.out.println("Celsius to Fahrenheit : " +
fahrenheit_to_celsius(far));

	}
}
```

Output:

```
Enter Celsius : 25
Celsius : 25.0
Celsius to Fahrenheit : 77.0
Enter Fahrenheit : 77
Fahrenheit : 77.0
Fahrenheit to Celsius : 25.0
```

Part 2: Classes and Methods

1. Create a Person class with the following properties: name, age, and gender. Implement methods to set and get these properties.

```
import java.util.Scanner;

class person{
	private String name;
    private int age;
    private String gender;

	public person(String name,int age,String gender){
		this.name = name;
		this.age = age;
```

```
      this.gender = gender;
}

public String get_name(){
      return name;
}

public int get_age(){
      return age;
}

public String get_gender(){
      return gender;
}

public void set_name(String name){
      this.name = name;
}

public void set_age(int age){
      this.age = age;
}

public void set_gender(String gender){
      this.gender = gender;
}

public static void main(String args[]){
      String nam;
      int agee;
      String genderr;

      Scanner s1 = new Scanner(System.in);

      System.out.print("Enter ur name : ");
      String u_name = s1.next();

      System.out.print("Enter ur age : ");
      int u_age = s1.nextInt();

      System.out.print("Enter ur gender : ");
      String u_gender = s1.next();

      person p1 = new person(u_name,u_age,u_gender);

      System.out.println("Name : " + p1.get_name());
      System.out.println("Age : " + p1.get_age());
      System.out.println("Gender : " + p1.get_gender());


      //set method
```

```
            p1.set_name(" J D");
            p1.set_age(23);
            p1.set_gender("male");

            System.out.println();

            System.out.println("Name : " + p1.get_name());
            System.out.println("Age : " + p1.get_age());
            System.out.println("Gender : " + p1.get_gender());
      }
}
OUTPUT:

Name : kaushal
Age : 30
Gender : male

Name : J D
Age : 23
Gender : male
```

2. Write a BankAccount class with the following properties: accountNumber, balance,
and owner. Implement methods to deposit, withdraw, and check the account balance.

```
import java.math.BigInteger;

class bac {
    private BigInteger an;
    private double bal;
    private String own;

    public bac(double acc_num, int b, String o) {
        this.an = BigInteger.valueOf((long) acc_num);
        this.bal = b;
        this.own = o;
    }

    public BigInteger get_an() {
        return an;
    }

    public String get_own() {
        return own;
    }

    public void dep(double amt) {
```

```java
        if (amt > 0) {
            bal += amt;
            System.out.println("Deposited amount: " + amt);
        } else {
            System.out.println("Deposited amount must be positive....");
        }
    }

    public void with(double amt) {
        if (amt > 0 && amt <= bal) {
            bal -= amt;
            System.out.println("Withdrawn amount: " + amt);
        } else {
            System.out.println("Invalid withdrawal amount");
        }
    }

    public void chk_bal() {
        System.out.println("Account Balance: " + bal);
    }

    public static void main(String[] args) {
        bac acc_1 = new bac(1987856001, 15000, "Jay");

        System.out.println();

        // Details of holder
        System.out.println("Name of the holder: " + acc_1.get_own());
        System.out.println("Number of the account: " + acc_1.get_an());
        acc_1.chk_bal();

        // Deposit money
        System.out.println();
        acc_1.dep(2222.27);
        acc_1.chk_bal();

        // Withdraw money
        System.out.println();
        acc_1.with(7222.27);
        acc_1.chk_bal();
    }
}
```

OUTPUT:

```
Name of the holder: Jay
Number of the account: 1987856001
Account Balance: 15000.0

Deposited amount: 2222.27
Account Balance: 17222.27

Withdrawn amount: 7222.27
Account Balance: 10000.0
```

3. Create a Circle class with the following properties: radius. Implement methods to calculate the area and circumference of the circle.

```
import java.util.Scanner;

class cir {
    private int r;

    public cir(int rad) {
        this.r = rad;
    }

    public void ar() {
        System.out.println("Area of circle: " + 3.14 * r * r);
    }

    public void pe() {
        System.out.println("Perimeter of circle: " + 2 * 3.14 * r);
    }

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

        System.out.println("Enter radius: ");
        int rad = s.nextInt();

        cir c = new cir(rad);

        c.ar();
        c.pe();
    }
}
Output:
Enter radius:
5

Area of circle: 78.5
Perimeter of circle: 31.400000000000002
```

Part 3: Conditional Statements and Loops

**1. Write a program that takes a number as input and determines whether it is even or odd
using an if-else statement.**

```
import java.util.Scanner;

public class Q1{
      public static void main(String args[])
      {
            System.out.print("Enter Number To Check Number Is Odd or
Even: ");

            Scanner sc = new Scanner(System.in);
            int n1 = sc.nextInt();

            if(n1 % 2 == 0)
            {
                  System.out.print("Number Is Even..!");
            }
            else
            {
                  System.out.print("Number Is Odd..!");
            }
      }outpuit:
Enter Number To Check Number Is Odd or Even:
7

Number Is Odd..!
```

**2. Implement a program that uses a switch-case statement to determine the day of the week
based on a numeric input (1 for Monday, 2 for Tuesday, etc.).**

```
import java.util.Scanner;

public class Q2{
      public static void main(String args[])
      {
            Scanner sc = new Scanner(System.in);
            int choice = sc.nextInt();

            switch(choice)
            {
                  case 1:
                  System.out.print("Monday..!");
                  break;

                  case 2:
                  System.out.print("Tuesday..!");
                  break;

                  case 3:
                  System.out.print("Wednesday..!");
                  break;

                  case 4:
                  System.out.print("Thursday..!");
                  break;

                  case 5:
                  System.out.print("Friday..!");
                  break;

                  case 6:
                  System.out.print("Saturday..!");
                  break;

                  case 7:
                  System.out.print("Sunday..!");
                  break;

                  default:
                  System.out.print("Invalid Choice.!");
            }
      }
}
Output:
1


Monday..!
```

3. Create a program that uses a for loop to print the first 10 multiples of a given number.

```
import java.util.Scanner;

public class Q3{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
```

```
            System.out.println("Enter Number: ");

            int n1 = sc.nextInt();
            int n2 = sc.nextInt();

            for(int i = 1;i <= 10;i++)
            {
                  System.out.println(n1 + "X" + i + " = " + n1*i);
            }

      }
}
```

```
Output:
Enter Number:
5

5X1 = 5
5X2 = 10
5X3 = 15
5X4 = 20
5X5 = 25
5X6 = 30
5X7 = 35
5X8 = 40
5X9 = 45
5X10 = 50
```

4. Write a program that uses a while loop to calculate the factorial of a given number.

```
import java.util.Scanner;
public class Q4{
      public static void main(String args[])
      {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter Number To Get Factorial: ");
            int number = sc.nextInt();
            int i = 1;
            int factorial = 1;

            while(i <= number)
            {
                  factorial *= i;
                  i++;
            }

            System.out.print("Factorial Of "+ number + " Is " +
factorial);
```

```
      }
}
Output:
Enter Number To Get Factorial:
5

Factorial Of 5 Is 120
```

Part 4: Putting It All Together

Create a Student class with the following properties: name, grade, and age.
Implement the following
methods:
setName(String name): Sets the student's name.
getGrade(): Returns the student's grade.
incrementAge(): Increments the student's age by 1.
printStudentInfo(): Prints the student's name, grade, and age.
In the main method, create an array of Student objects and perform the following tasks:
Prompt the user to enter the number of students.
Create the Student objects and populate their information.
Use a for loop to iterate through the array and call the printStudentInfo() method for each
student.
Find the student with the highest grade and print their information.

```
import java.util.Scanner;

class stu {
    private String n; // Name of studnt
    private int g;    // Grade of studnt
    private int a;    // Age of studnt

    // Method for set name of studnt
    public void set_n(String name) {
        this.n = name;
    }

    // Method for get grade of studnt
    public int get_g() {
        return g;
    }

    // Method for increment age of studnt by one
    public void inc_a(int age) {
        this.a = age + 1; // Increment age by one
    }

    // Method for print info of studnt
    public void print() {
        System.out.println("Name: " + n);
        System.out.println("Grade: " + g);
        System.out.println("Age: " + a);
    }

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

        // Input number of studnt
        System.out.print("Enter the number of students: ");
        int num = s.nextInt();
        s.nextLine(); // Consume the new line charcter

        stu[] stus = new stu[num]; // Array for holding the studnt objects

        // Loop for input details of each studnt
        for (int i = 0; i < num; i++) {
            stus[i] = new stu(); // Create new studnt object
            System.out.print("Enter studnt " + (i + 1) + " name: ");
            stus[i].set_n(s.nextLine());
            System.out.print("Enter studnt " + (i + 1) + " grade: ");
            stus[i].g = s.nextInt(); // Directly assing grade
            System.out.print("Enter studnt " + (i + 1) + " age: ");
            stus[i].inc_a(s.nextInt()); // Increment age of studnt
            s.nextLine(); // Consume the new line charcter
        }

        // Print all info of studnt
        for (stu student : stus) {
            student.print();
        }

        // Find studnt with the highest grade
        stu highest = stus[0];
        for (int i = 1; i < num; i++) {
            if (stus[i].get_g() > highest.get_g()) {
                highest = stus[i]; // Update if current studnt has more
higher grade
            }
        }

        // Print info of studnt with highest grade
        System.out.println("Studnt with highest grade is:");
        highest.print();

        s.close(); // Close the scanner
    }
}
```

Output:

```
Enter the number of students:
3
Enter student 1 name:
Jay
Enter student 1 grade:
85
Enter student 1 age:
20
Enter student 2 name:
Kausal
Enter student 2 grade:
90
Enter student 2 age:
21
Enter student 3 name:
Tirth
Enter student 3 grade:
88
Enter student 3 age:
22


Name: Jay
Grade: 85
Age: 21
Name: Kausal
Grade: 90
Age: 22
Name: Tirth
Grade: 88
Age: 23
Studnt with highest grade is:
Name: Kausal
Grade: 90
Age: 22
```

Problem 1: Currency Converter
Write a program that can convert between different currencies. The program should have the
following features:
Prompt the user to enter the amount they want to convert.
Provide a list of available currencies (e.g., USD, EUR, JPY, GBP, INR) and ask the user to select
the source and target currencies.
Implement the currency conversion logic using appropriate exchange rates.

Display the converted amount to the user.

```
import java.util.Scanner;

class currency_converter{
      public static double get_exchange_rate(String source,String
target){
            double exchange_rate = 1.0;


            switch(source) {
                  case "USD":
                        switch(target) {
                              case "EUR": exchange_rate = 0.85;
                              break;

                              case "JPY": exchange_rate = 110.0;
                              break;

                              case "GBP": exchange_rate = 0.75;
                              break;

                              case "INR": exchange_rate = 74.0;
                              break;

                              default:
                                    System.out.println("Target currency
not supported ");
                                    exchange_rate = -1; // set to -1
                                    break;
                        }
                        break;

                  case "EUR":
                        switch(target) {
                              case "USD": exchange_rate = 1.18;
                              break;

                              case "JPY": exchange_rate = 129.53;
                              break;

                              case "GBP": exchange_rate = 0.88;
                              break;

                              case "INR": exchange_rate = 87.0;
                              break;

                              default:
                                    System.out.println("Target currency
not supported");
```

```
                                   exchange_rate = -1;
                                   break;
                       }
                       break;

                 case "JPY":
                       switch(target) {
                             case "USD": exchange_rate = 0.0091;
                             break;

                             case "EUR": exchange_rate = 0.0077;
                             break;

                             case "GBP": exchange_rate = 0.0068;
                             break;

                             case "INR": exchange_rate = 0.67;
                             break;

                             default:
                                   System.out.println("Target currency
not supported ");
                                   exchange_rate = -1;
                                   break;
                       }
                       break;

                 case "GBP":
                       switch(target) {
                             case "USD": exchange_rate = 1.33;
                             break;

                             case "EUR": exchange_rate = 1.14;
                             break;

                             case "JPY": exchange_rate = 151.0;
                             break;

                             case "INR": exchange_rate = 98.0;
                             break;

                             default:
                                   System.out.println("Target currency
not supported ");
                                   exchange_rate = -1;
                                   break;
                       }
                       break;

                 case "INR":
                       switch(target) {
                             case "USD": exchange_rate = 0.014;
                             break;

                             case "EUR": exchange_rate = 0.011;
                             break;

                             case "JPY": exchange_rate = 1.49;
                             break;

                             case "GBP": exchange_rate = 0.010;
                             break;

                             default:
                                   System.out.println("Target currency
not supported ");
                                   exchange_rate = -1;
                                   break;
                       }
                       break;

                 default:
                       System.out.println("Source currency not
supported");
                       exchange_rate = -1;
                       break;
             }

              return exchange_rate;
        }

        public static void main(String args[]){
             Scanner s1 = new Scanner(System.in);

             System.out.print("Enter the amount you want to convert : ");
          double amount = s1.nextDouble();

             String currencies[] = {"USD", "EUR", "JPY", "GBP", "INR"};


             System.out.println();
             System.out.println("Select the source currency : ");
             for(int i=0 ; i < currencies.length ; i++){
                 System.out.println((i+1) + "." + currencies[i]);

             }

             int s_index = s1.nextInt()-1;
             String source_currency_index = currencies[s_index];


             System.out.println();
             System.out.println("Select the target currency : ");
             for(int i=0 ; i < currencies.length ; i++){
                 System.out.println((i+1) + "." + currencies[i]);

             }

             int t_index = s1.nextInt()-1;
             String target_currency_index = currencies[t_index];

             double ex_rate =
get_exchange_rate(source_currency_index,target_currency_index);
             double calculate_amount = amount * ex_rate;


             System.out.println();
             System.out.printf("Converted Amount : %.2f %s" ,
calculate_amount, target_currency_index);


        }
}
Output:
Enter the amount you want to convert :
100

Select the source currency :
1.USD
2.EUR
3.JPY
4.GBP
5.INR
1

Select the target currency :
1.USD
2.EUR
3.JPY
4.GBP
5.INR
2

Converted Amount : 85.00 EUR
```

Problem 2: Temperature Analyzer
Create a program that analyzes temperature data. The program should have the following features:
Allow the user to input a series of temperature readings (in Celsius).

Calculate and display the average temperature.

Determine and display the highest and lowest temperatures.

Categorize the temperatures into "cold," "mild," and "hot" based on predefined temperature ranges.

```
import java.util.Scanner;

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

        System.out.print("Enter the number of temperature readings: ");
        int numReadings = scanner.nextInt();

        double[] temperatures = new double[numReadings];
        double sum = 0;
        double highestTemp = Double.MIN_VALUE;
        double lowestTemp = Double.MAX_VALUE;

        for (int i = 0; i < numReadings; i++) {
            System.out.print("Enter temperature in Celsius: ");
            temperatures[i] = scanner.nextDouble();
            sum += temperatures[i];
            if (temperatures[i] > highestTemp) {
                highestTemp = temperatures[i];
            }
            if (temperatures[i] < lowestTemp) {
                lowestTemp = temperatures[i];
            }
        }

        double averageTemp = sum / numReadings;

        System.out.println("Average Temperature: " + averageTemp);
        System.out.println("Highest Temperature: " + highestTemp);
        System.out.println("Lowest Temperature: " + lowestTemp);

        for (double temp : temperatures) {
            if (temp < 15) {
                System.out.println(temp + "°C is cold.");
            } else if (temp >= 15 && temp < 30) {
                System.out.println(temp + "°C is mild.");
            } else {
                System.out.println(temp + "°C is hot.");
            }
        }

        scanner.close();
    }
}
Output:
Enter the number of temperature readings: 5
Enter temperature in Celsius: 10
Enter temperature in Celsius: 20
Enter temperature in Celsius: 30
Enter temperature in Celsius: 15
Enter temperature in Celsius: 25

Average Temperature: 20.0
Highest Temperature: 30.0
Lowest Temperature: 10.0
10.0°C is cold.
20.0°C is mild.
30.0°C is hot.
15.0°C is mild.
25.0°C is hot.
```

Problem 3: Student Grade Management
Develop a program to manage student grades. The program should have the following features:
Create a Student class with properties for name, student ID, and grades (in an array).
Implement methods to:
Add a new student
Update a student's grades
Calculate the average grade for a student
Find the student with the highest average grade
Prompt the user to perform various operations, such as adding a new student, updating a
student's grades, and displaying the student with the highest average grade.

```
import java.util.Scanner;

class stu {
    private String n; // Name of studnt
    private String sid; // Student ID
    private int[] g; // Array for grades
    private int gc; // to keep track of number of grades added

    public stu(String name, String sid) {
        this.n = name;
        this.sid = sid;
        this.g = new int[5]; // initial size for grades array
        this.gc = 0; // Initialize grade count
    }

    public String get_n() {
        return n; // Return name of studnt
    }

    public String getsid() {
        return sid; // Return student ID
    }

    public void add_g(int grade) {
        if (gc < g.length) {
            g[gc] = grade;
            gc++;
        } else {
            System.out.println("Cannot add more grades. Grade array is
full.");
        }
    }

    public void update_g(int[] grades) {
        this.g = grades; // Update grades
        this.gc = grades.length; // Update grade count
    }

    public double avg() {
        if (gc == 0) {
            return 0; // Return 0 if no grades
        }
        int sum = 0;
        for (int i = 0; i < gc; i++) {
            sum += g[i]; // Sum up grades
        }
        return (double) sum / gc; // Calculate average
    }
}

class stu_manager {
    private stu[] stus; // Array for holding studnts
    private int sc; // Student count

    public stu_manager() {
        this.stus = new stu[10]; // initial size for studnts array
        this.sc = 0; // Initialize student count
    }

    public void add_s() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter studnt's name: ");
        String name = scanner.nextLine();
        System.out.print("Enter studnt ID: ");
        String sid = scanner.nextLine();
        stu student = new stu(name, sid);

        if (sc < stus.length) {
            stus[sc] = student; // Add new studnt
            sc++;
            System.out.println("Studnt added successfully.");
        } else {
            System.out.println("Cannot add more studnts. Studnt array is
full.");
        }
    }

    public void update_g() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter studnt ID to update grades: ");
        String sid = scanner.nextLine();

        stu s_update = find_sid(sid);

        if (s_update == null) { // If studnt not found
            System.out.println("Studnt not found.");
            return;
        }

        int[] grades = new int[5]; // Array to hold grades for 5 subjects
        String[] subjects = {"Math", "Science", "English", "History", "Art"};

        for (int i = 0; i < subjects.length; i++) {
            while (true) {
                try {
                    System.out.printf("Enter grade for %s: ", subjects[i]);
                    grades[i] = Integer.parseInt(scanner.nextLine().trim());

                    if (grades[i] < 0 || grades[i] > 100) { // Check grade
range
                        System.out.println("Please enter a grade between 0
and 100.");
                    } else {
                        break; // Break if valid grade
                    }
                } catch (NumberFormatException e) {
                    System.out.println("Invalid input. Please enter a numeric
grade.");
                }
            }
        }

        s_update.update_g(grades); // Update grades of studnt
        System.out.println("Grades updated successfully.");
    }

    public void display_highest() {
        if (sc == 0) { // Check if no studnts
```

```
            System.out.println("No studnts to display.");
            return;
        }

        stu highest_avg_stu = stus[0]; // Assume first is highest
        double highest_avg = highest_avg_stu.avg(); // Get average of first

        for (int i = 1; i < sc; i++) {
            double current_avg = stus[i].avg(); // Get current studnt average
            if (current_avg > highest_avg) { // If current is higher
                highest_avg = current_avg;
                highest_avg_stu = stus[i]; // Update highest
            }
        }

        System.out.println("Studnt with highest average grade:");
        System.out.println("Name: " + highest_avg_stu.get_n());
        System.out.println("Studnt ID: " + highest_avg_stu.getsid());
        System.out.println("Average Grade: " + highest_avg);
    }

    private stu find_sid(String sid) {
        for (int i = 0; i < sc; i++) {
            if (stus[i].getsid().equals(sid)) { // Check if IDs match
                return stus[i]; // Return studnt
            }
        }
        return null; // Not found
    }

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

        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Add a new studnt");
            System.out.println("2. Update studnt's grades");
            System.out.println("3. Display studnt with highest average
grade");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");

            int choice = scanner.nextInt();
            scanner.nextLine(); // consume newline

            switch (choice) {
                case 1:
                    manager.add_s(); // Add new studnt
                    break;
                case 2:
                    manager.update_g(); // Update grades
                    break;
                case 3:
                    manager.display_highest(); // Display highest
                    break;
                case 4:
                    System.out.println("Exiting program.");
                    return; // Exit program
                default:
                    System.out.println("Invalid choice. Please enter a number
between 1 and 4.");
            }
        }
    }
}
```

Output:

```
Menu:
1. Add a new student
2. Update student's grades
3. Display student with highest average grade
4. Exit
Enter your choice: 1
Enter student's name: Jay
Enter student ID: 001
Student added successfully.

Menu:
1. Add a new student
2. Update student's grades
3. Display student with highest average grade
4. Exit
Enter your choice: 1
Enter student's name: Tirth
Enter student ID: 002
Student added successfully.

Menu:
1. Add a new student
2. Update student's grades
3. Display student with highest average grade
4. Exit
Enter your choice: 2
Enter student ID to update grades: 001
Enter grade for Math: 85
Enter grade for Science: 90
Enter grade for English: 78
Enter grade for History: 88
Enter grade for Art: 92
Grades updated successfully.

Menu:
1. Add a new student
2. Update student's grades
3. Display student with highest average grade
```

```
4. Exit
Enter your choice: 2
Enter student ID to update grades: 002
Enter grade for Math: 70
Enter grade for Science: 75
Enter grade for English: 80
Enter grade for History: 90
Enter grade for Art: 85
Grades updated successfully.

Menu:
1. Add a new student
2. Update student's grades
3. Display student with highest average grade
4. Exit
Enter your choice: 3
Student with highest average grade:
Name: Jay
Student ID: 001
Average Grade: 86.6

Menu:
1. Add a new student
2. Update student's grades
3. Display student with highest average grade
4. Exit
Enter your choice: 4
Exiting program.
```

Problem 4: Fibonacci Sequence Generator
Write a program that generates the Fibonacci sequence. The program should have the following
features:
Prompt the user to enter the number of Fibonacci numbers to generate.
Use a loop to calculate and display the Fibonacci sequence up to the specified number of
terms.
Optionally, allow the user to choose whether to display the sequence using
a for loop, while loop, or recursive method.

```
import java.util.Scanner;

class fib {

    public static void for_l(int n) {
        int a = 0, b = 1, c;
        for (int i = 0; i < n; i++) {
            System.out.print(a + " ");
            c = a + b;
            a = b;
            b = c;
        }
        System.out.println();
    }

    public static void while_l(int n) {
        int a = 0, b = 1, c;
        int i = 0;
        while (i < n) {
            System.out.print(a + " ");
            c = a + b;
            a = b;
            b = c;
            i++;
        }
        System.out.println();
    }

    public static int fib_rec(int n) {
        if (n <= 1) {
            return n;
        } else {
            return fib_rec(n - 1) + fib_rec(n - 2);
        }
    }

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

        System.out.println("1. For Loop.");
        System.out.println("2. While Loop.");
        System.out.println("3. Recursive.");
        System.out.println("4. Exit.");
        int choice = sc.nextInt();

        switch (choice) {
            case 1:
                System.out.print("Enter the number of Fibbonacci numbers to
generate: ");
                int n = sc.nextInt();
                for_l(n);
                break;

            case 2:
                System.out.print("Enter the number of Fibbonacci numbers to
generate: ");
                int n1 = sc.nextInt();
                while_l(n1);
                break;
            case 3:
                System.out.print("Enter the number of Fibbonacci numbers to
generate: ");
                int n2 = sc.nextInt();
                for (int i = 0; i < n2; i++) {
                    System.out.print(fib_rec(i) + " ");
                }
                System.out.println();
                break;

            case 4:
                System.out.print("Exit..!");
                break;

            default:
                System.out.print("Invalid Choce..!");
        }
    }
}
```

Outpuit:

```
1. For Loop.
2. While Loop.
3. Recursive.
4. Exit.
Enter your choice: 1
Enter the number of Fibonacci numbers to generate: 5
0 1 1 2 3

1. For Loop.
2. While Loop.
3. Recursive.
4. Exit.
Enter your choice: 2
Enter the number of Fibonacci numbers to generate: 5
0 1 1 2 3

1. For Loop.
2. While Loop.
3. Recursive.
4. Exit.
Enter your choice: 3
Enter the number of Fibonacci numbers to generate: 5
0 1 1 2 3
```

Problem 5: Palindrome Checker

Develop a program that checks whether a given string is a palindrome. The program should have the following features:
Prompt the user to enter a string.
Implement a method to determine if the string is a palindrome (a word, phrase, number, or other sequence of characters that reads the same backward as forward).
Display the result (whether the input is a palindrome or not) to the user.

```
import java.util.Scanner;

 class PalindromeChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        if (isPalindrome(input)) {
            System.out.println("The input string is a palindrome.");
        } else {
            System.out.println("The input string is not a
palindrome.");
        }
    }

    public static boolean isPalindrome(String str) {
        int left = 0;
        int right = str.length() - 1;

        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
Output:
Enter a string: radar
The input string is a palindrome.
```

****

```
import java.util.Scanner;
import java.io.* ;

class palindrome{
      private int s;

      public palindrome(int n){
            this.s = n;
            }

      public void check_palindrome(int str){
            int s1 = 0;
            int temp = str;
            while (str > 0){
                  int remainder = str % 10;
                  s1 = (s1 * 10) + remainder;
                  str = Math.floorDiv(str,10);
            }
            if(temp == s1){
                  System.out.println("palindromw");

            }else{
                  System.out.println("not palindromw");
            }
      }


      public static void main(String args[]){
            Scanner s1 = new Scanner(System.in);

            System.out.println("Enter number : ");
            int rad = s1.nextInt();

            palindrome c = new palindrome(rad);

            c.check_palindrome(rad);
      }
}
```

Problem 6: BMI Calculator
Create a program that calculates a person's Body Mass Index (BMI) and provides an interpretation of
the result. The program should have the following features:
Prompt the user to enter their height (in meters) and weight (in kilograms).
Calculate the BMI using the formula: BMI = weight / (height * height).

Determine the BMI category (underweight, normal, overweight, or obese) based on the
calculated BMI.
Display the user's BMI value and the corresponding BMI category.

```
import java.util.Scanner;

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

            System.out.print("Enter Your Height: ");
            double height = sc.nextDouble();

            System.out.print("Enter Your Weight: ");
            double weight = sc.nextDouble();

             double BMICalculate = weight / (height * height);


             if(BMICalculate < 18.5)
             {
                   System.out.print("UnderWeight...!");
             }

             else if(BMICalculate > 18.5 && BMICalculate < 24.9)
             {
                   System.out.print("Normal...!");
             }

             else if(BMICalculate > 25 && BMICalculate < 30)
             {
                   System.out.print("Overweight...!");
             }

             else if(BMICalculate > 40)
             {
                   System.out.print("Obese..!");
             }
      }
}
```

```
Output:
Enter Your Height (in meters): 1.80
Enter Your Weight (in kilograms): 85
Your BMI is: 26.23
Overweight...

Enter Your Height (in meters): 1.75
Enter Your Weight (in kilograms): 95
```

```
Your BMI is: 31.02
Obese...

Enter Your Height (in meters): 1.60
Enter Your Weight (in kilograms): 50
Your BMI is: 19.53
Normal weight...
```

Problem 7: Leap Year Checker
Write a program that checks whether a given year is a leap year. The program should have the
following features:
Prompt the user to enter a year.
Implement the logic to determine if the year is a leap year (a year divisible by 4, except for
years divisible by 100, unless they are also divisible by 400).
Display whether the entered year is a leap year or not.

```
import java.util.Scanner;

class LeapYear{

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

            System.out.print("Enter a Year To Check Leap Year or Not: 
");
            int year = sc.nextInt();

            if (isLeapyear(year))
            {
                  System.out.print(year + " Is Leap Year..!");
            }
            else
            {
                  System.out.print(year + " Is Not Leap Year..!");
            }
      }
      public static boolean isLeapyear(int year)
      {
            if(year % 4 == 0)
            {
                  if(year % 100 == 0 && year % 400 == 0){
                  return true;
                  }
                  else if(year % 100 == 0)
                  {
                        return false;
                  }
                  else
                  {
                        return true;
                  }

                  return true;
            }
            else
            {
                  return false;
            }


      }
}
Output:
Enter a Year To Check Leap Year or Not: 2020
2020 Is Leap Year..!

Enter a Year To Check Leap Year or Not: 2021
2021 Is Not Leap Year..!
```

Problem 8: Coin Toss Simulator
Develop a program that simulates a coin toss. The program should have the following features:
Allow the user to choose the number of coin tosses to perform.
Randomly generate the outcome of each coin toss (heads or tails).
Keep track of the number of heads and tails.
Display the results, including the number of heads and tails, as well as the percentage of each outcome.

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

class hello {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Random r1 = new Random();

        System.out.print("Enter the number of coin tosses : ");
        int num_tosses = scanner.nextInt();

        int heads_count = 0; //  for heads
        int tails_count = 0; //  for tails

        for (int i = 0; i < num_tosses; i++) {
            boolean is_heads = r1.nextBoolean(); //randomly get true or
false value for hhead and tail
            if (is_heads) {
                heads_count++; // increment heads
            } else {
                tails_count++;
            }
        }

        double heads_percentage = ((double) heads_count / num_tosses) *
100;
        double tails_percentage = ((double) tails_count / num_tosses) *
100;

        System.out.println("Results of " + num_tosses + " coin
tosses:");
        System.out.println("Heads: " + heads_count + " (" +
String.format("%.2f", heads_percentage) + "%)");
        System.out.println("Tails: " + tails_count + " (" +
String.format("%.2f", tails_percentage) + "%)");

    }
}
Output:
Enter the number of coin tosses: 100
Results of 100 coin tosses:
Heads: 49 (49.00%)
Tails: 51 (51.00%)

Enter the number of coin tosses: 50
Results of 50 coin tosses:
Heads: 24 (48.00%)
Tails: 26 (52.00%)
```

Problem 9: Password Validator

Create a program that validates the strength of a user's password. The program should have the following features:

Prompt the user to enter a password.

Implement a set of rules to check the password's strength, such as minimum length, inclusion of uppercase and lowercase letters, digits, and special characters.

Provide feedback to the user on the password's strength (e.g., weak, medium, strong).

Allow the user to try again if the password is not strong enough.

```
import java.util.Scanner;
public class PasswordValidator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.print("Enter a password: ");
            String password = scanner.nextLine();

            String feedback = validatePassword(password);

            System.out.println(feedback);

            if ("Strong password!".equals(feedback)) {
                break;
            } else {
                System.out.println("Please try again.");
            }
        }

        scanner.close();
    }

    public static String validatePassword(String password) {
        if (password.length() < 8) {
            return "Password is too short. It must be at least 8
characters long.";
        }

        boolean hasUppercase = false;
        boolean hasLowercase = false;
        boolean hasDigit = false;
        boolean hasSpecialChar = false;

        for (char c : password.toCharArray()) {
            if (Character.isUpperCase(c)) {
                hasUppercase = true;
            } else if (Character.isLowerCase(c)) {
                hasLowercase = true;
            } else if (Character.isDigit(c)) {
                hasDigit = true;
            } else if (!Character.isLetterOrDigit(c)) {
                hasSpecialChar = true;
            }
        }

        if (!hasUppercase) {
            return "Password must include at least one uppercase 
letter.";
        }

        if (!hasLowercase) {
            return "Password must include at least one lowercase 
letter.";
        }

        if (!hasDigit) {
            return "Password must include at least one digit.";
        }

        if (!hasSpecialChar) {
            return "Password must include at least one special 
character.";
        }

        return "Strong password!";
    }
}
```

Output:

```
Enter a password: abc123
Password is too short. It must be at least 8 characters long.
Please try again.

Enter a password: abcdefgH
Password must include at least one digit.
Please try again.

Enter a password: abc12345
Password must include at least one uppercase letter.
Please try again.
```

```
Enter a password: Abc12345!
Strong password!

Enter a password: Strong@Password1
Strong password!
```

Problem 10: Grocery List Manager

Develop a program that helps users manage their grocery list. The program should have the

following features:

Allow the user to add items to the grocery list.

Provide the ability to mark items as purchased or remove them from the list.

Display the current grocery list, including the purchased and remaining items.

Implement a search functionality to find specific items on the list.

Optionally, allow the user to save and load the grocery list to/from a file.

```
import java.util.Scanner;

 class GlistManager {
    private static String[] glist = new String[10];
    private static int n = 0;

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

        while (flagg) {
            System.out.println("1. Add item to the list");
            System.out.println("2. Mark item as purchased");
            System.out.println("3. Remove item from the list");
            System.out.println("4. Display grocery list");
            System.out.println("5. Search for an item");
            System.out.println("6. Exit");

            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // consumme newline

            switch (choice) {
                case 1:
                    add_item(scanner);
                    break;
                case 2:
                    mark_purchased(scanner);
                    break;
                case 3:
                    remove_item(scanner);
                    break;
                case 4:
                    displaylist();
                    break;
                case 5:
                    searchitem(scanner);
                    break;
                case 6:
                    flagg = false;
                    break;
                default:
                    System.out.println("Invalid choice. Please try
again.");
            }
        }
        scanner.close();
    }

    private static void add_item(Scanner scanner) {
        System.out.print("Enter the item to add: ");
        String item = scanner.nextLine();
        glist[n] = item;
        n++;
        System.out.println("Item added to the list.");
    }

    private static void mark_purchased(Scanner scanner) {
        System.out.print("Enter the item to mark as purchased: ");
        String item = scanner.nextLine();
        for (int i = 0; i < n; i++) {
            if (glist[i].equals(item)) {
                glist[i] = "Purchased - " + item;
                System.out.println("Item marked as purchased.");
                return;
            }
        }
        System.out.println("Item not found in the list.");
    }

    private static void remove_item(Scanner scanner) {
        System.out.print("Enter the item to remove: ");
        String item = scanner.nextLine();
        for (int i = 0; i < n; i++) {
            if (glist[i].equals(item)) {
                for (int j = i; j < n - 1; j++) {
                    glist[j] = glist[j + 1];
                }
                n--;
                System.out.println("Item removed from the list.");
                return;
            }
        }
        System.out.println("Item not found in the list.");
    }

    private static void displaylist() {
        System.out.println("Grocery List:");
        for (int i = 0; i < n; i++) {
            System.out.println((i + 1) + ". " + glist[i]);
        }
    }

    private static void searchitem(Scanner scanner) {
        System.out.print("Enter the item to search for: ");
        String item = scanner.nextLine();
        for (int i = 0; i < n; i++) {
            if (glist[i].contains(item)) {
                System.out.println("Item found in the list: " +
glist[i]);
                return;
            }
        }
        System.out.println("Item not found in the list.");
    }
}
```

Output:

```
1. Add item to the list
2. Mark item as purchased
3. Remove item from the list
4. Display grocery list
5. Search for an item
6. Exit
Enter your choice: 1
Enter the item to add: Milk
Item added to the list.

1. Add item to the list
2. Mark item as purchased
3. Remove item from the list
4. Display grocery list
5. Search for an item
6. Exit
Enter your choice: 1
Enter the item to add: Eggs
Item added to the list.

1. Add item to the list
2. Mark item as purchased
3. Remove item from the list
4. Display grocery list
```

```
5. Search for an item
6. Exit
Enter your choice: 1
Enter the item to add: Bread
Item added to the list.

1. Add item to the list
2. Mark item as purchased
3. Remove item from the list
4. Display grocery list
5. Search for an item
6. Exit
Enter your choice: 2
Enter the item to mark as purchased: Milk
Item marked as purchased.

1. Add item to the list
2. Mark item as purchased
3. Remove item from the list
4. Display grocery list
5. Search for an item
6. Exit
Enter your choice: 2
Enter the item to mark as purchased: Bread
Item marked as purchased.

1. Add item to the list
2. Mark item as purchased
3. Remove item from the list
4. Display grocery list
5. Search for an item
6. Exit
Enter your choice: 4
Grocery List:
1. Purchased - Milk
2. Eggs
3. Purchased - Bread

1. Add item to the list
2. Mark item as purchased
3. Remove item from the list
4. Display grocery list
5. Search for an item
6. Exit
Enter your choice: 5
Enter the item to search for: Milk
Item found in the list: Purchased - Milk

1. Add item to the list
2. Mark item as purchased
3. Remove item from the list
4. Display grocery list
5. Search for an item
```

```
6. Exit
Enter your choice: 3
Enter the item to remove: Eggs
Item removed from the list.

1. Add item to the list
2. Mark item as purchased
3. Remove item from the list
4. Display grocery list
5. Search for an item
6. Exit
Enter your choice: 4
Grocery List:
1. Purchased - Milk
2. Purchased - Bread

1. Add item to the list
2. Mark item as purchased
3. Remove item from the list
4. Display grocery list
5. Search for an item
6. Exit
Enter your choice: 6
```