1.  a) Write a Java program that prompts the user to enter an integer, reads the input, and displays the entered integer on the console.

b) Develop a Java program that reads two floating-point numbers from the user, calculates their average, and displays the result on the console with two decimal places.

import java.util.Scanner;

public class ReadInteger {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter an integer: ");

int number = scanner.nextInt();

System.out.println("The entered integer is: " + number);

scanner.close();

}

}

import java.util.Scanner;

public class CalculateAverage {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter first floating-point number: ");

double num1 = scanner.nextDouble();

System.out.print("Enter second floating-point number: ");

double num2 = scanner.nextDouble();

double average = (num1 + num2) / 2;

System.out.printf("The average is: %.2f%n", average);

scanner.close();

}

}

2.

Implement a Java program that simulates a basic calculator with functionalities to perform addition, subtraction, multiplication, and division.  The program should prompt the user to enter two numbers and an operator (+, -, , /), perform the corresponding operation, and display the result.  Ensure to handle division by zero and invalid operator inputs.

import java.util.Scanner;

public class BasicCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

double num1 = scanner.nextDouble();

System.out.print("Enter operator (+, -, \*, /): ");

char operator = scanner.next().charAt(0);

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

double num2 = scanner.nextDouble();

double result = 0;

switch (operator) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

if (num2 == 0) {

System.out.println("Error: Division by zero");

} else {

result = num1 / num2;

}

break;

default:

System.out.println("Invalid operator");

return; // Exit the program if the operator is invalid

}

System.out.println(num1 + " " + operator + " " + num2 + " = " + result);

}

}

## 3. Write an Java program to determine if a number n is happy.  A happy number is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits. Repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1.Those numbers for which this process ends in 1 are happy. Print true if n is a happy number, and false if not

import java.util.HashSet;

import java.util.Scanner;

public class HappyNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

System.out.println(isHappy(n));

}

public static boolean isHappy(int n) {

HashSet<Integer> seen = new HashSet<>();

while (n != 1 && !seen.contains(n)) {

seen.add(n);

n = sumOfSquaresOfDigits(n);

}

return n == 1;

}

public static int sumOfSquaresOfDigits(int n) {

int sum = 0;

int digit;

while (n != 0) {

digit = n % 10;

sum += digit \* digit;

n