import java.util.Scanner;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Задание 1: Проверка возможности создания треугольника  
 System.*out*.println("Task 1: Checking the possibility of creating a triangle");  
 System.*out*.print("Enter the length of the first segment: ");  
 double a = scanner.nextDouble();  
 System.*out*.print("Enter the length of the second segment: ");  
 double b = scanner.nextDouble();  
 System.*out*.print("Enter the length of the third segment: ");  
 double c = scanner.nextDouble();  
 boolean isTriangle = *triangle*(a, b, c);  
 System.*out*.println("Is it possible to make a triangle: " + isTriangle);  
  
 // Задание 2: Расстояние между двумя точками  
 System.*out*.println("\nTask 2: The distance between two points");  
 System.*out*.print("Enter the x1 coordinate: ");  
 double x1 = scanner.nextDouble();  
 System.*out*.print("Enter the y1 coordinate: ");  
 double y1 = scanner.nextDouble();  
 System.*out*.print("Enter the x2 coordinate: ");  
 double x2 = scanner.nextDouble();  
 System.*out*.print("Enter the y2 coordinate: ");  
 double y2 = scanner.nextDouble();  
 double distance = *distance*(x1, y1, x2, y2);  
 System.*out*.println("Distance between points: " + distance);  
  
 // Задание 3: Возведение в степень (итеративный подход)  
 System.*out*.println("\nTask 3: Exponentiation (iterative approach)");  
 System.*out*.print("Enter the base of the degree (a valid positive number): ");  
 double a3 = scanner.nextDouble();  
 System.*out*.print("Enter the exponent (integer): ");  
 int n3 = scanner.nextInt();  
 double result3 = *powerLoop*(a3, n3);  
 System.*out*.println("Result: " + result3);  
  
 // Задание 4: Возведение в степень (рекурсивный подход)  
 System.*out*.println("\nTask 4: Exponentiation (recursive approach)");  
 System.*out*.print("Enter the base of the degree (a valid positive number): ");  
 double a4 = scanner.nextDouble();  
 System.*out*.print("Enter the exponent (a non-negative integer): ");  
 int n4 = scanner.nextInt();  
 double result4 = *power*(a4, n4);  
 System.*out*.println("Result: " + result4);  
  
 // Задание 5: Вычисление числа Трибоначчи  
 System.*out*.println("\nTask 5: Calculating the Fibonacci number");  
 System.*out*.print("Enter the number of the Tribonacci number: ");  
 int n5 = scanner.nextInt();  
 int tribonacciNumber = *tribonacci*(n5);  
 System.*out*.println("Number of the Tribonacci: " + tribonacciNumber);  
 }  
  
 // Задание 1: Проверка возможности создания треугольника  
 public static boolean triangle(double a, double b, double c) {  
 if (a <= 0 || b <= 0 || c <= 0) {  
 return false; // Если хотя бы одна сторона меньше или равна нулю, треугольник невозможен  
 }  
 return (a + b > c && a + c > b && b + c > a); // Проверка неравенств треугольника  
 }  
  
 // Задание 2: Расстояние между двумя точками  
 public static double distance(double x1, double y1, double x2, double y2) {  
 return Math.*sqrt*(Math.*pow*(x2 - x1, 2) + Math.*pow*(y2 - y1, 2)); // Формула расстояния по теореме Пифагора  
 }  
  
 // Задание 3: Возведение в степень (итеративный подход)  
 public static double powerLoop(double a, int n) {  
 if (a <= 0) {  
 return Double.*NaN*; // Возвращаем NaN для некорректного основания  
 }  
 if (n == 0) {  
 return 1; // Любое число в степени 0 равно 1  
 }  
 double result = 1;  
 if (n > 0) {  
 for (int i = 0; i < n; i++) {  
 result \*= a; // Умножаем результат на основание n раз  
 }  
 } else {  
 for (int i = 0; i > n; i--) {  
 result /= a; // Делим результат на основание n раз  
 }  
 }  
 return result;  
 }  
  
 // Задание 4: Возведение в степень (рекурсивный подход)  
 public static double power(double a, int n) {  
 if (a <= 0) {  
 return Double.*NaN*; // Возвращаем NaN для некорректного основания  
 }  
 if (n == 0) {  
 return 1; // Любое число в степени 0 равно 1  
 }  
 if (n > 0) {  
 return a \* *power*(a, n - 1); // Рекурсивный вызов с уменьшением показателя степени  
 } else {  
 return 1 / *power*(a, -n); // Обработка отрицательной степени  
 }  
 }  
  
 // Задание 5: Вычисление числа Трибоначчи  
 public static int tribonacci(int n) {  
 if (n < 0) {  
 return -1; // Обработка некорректных входных данных  
 }  
 if (n == 0 || n == 1) {  
 return 0; // Базовые случаи  
 }  
 if (n == 2) {  
 return 1; // Базовый случай  
 }  
 return *tribonacci*(n - 1) + *tribonacci*(n - 2) + *tribonacci*(n - 3); // Рекурсивный вызов  
 }  
}

