1. **Write a Java program that prints "Hello, World!" to the console.**

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

1. **Write a Java program that prints your name and age on separate lines.**

public class PrintNameAge {

public static void main(String[] args) {

String name = "John Doe";

int age = 25;

System.out.println("Name: " + name);

System.out.println("Age: " + age);

}

}

1. **Write a Java program to add two integers and print the result.**

public class AddTwoIntegers {

public static void main(String[] args) {

int num1 = 10;

int num2 = 20;

int sum = num1 + num2;

System.out.println("Sum: " + sum);

}

}

1. **Write a Java program to find the product of two floating-point numbers.**

public class ProductOfTwoFloats {

public static void main(String[] args) {

float num1 = 5.5f;

float num2 = 4.5f;

float product = num1 \* num2;

System.out.println("Product: " + product);

}

}

1. **Write a Java program to declare multiple variables of different data types and print their values.**

public class MultipleVariables {

public static void main(String[] args) {

int intVar = 100;

float floatVar = 10.5f;

char charVar = 'A';

String stringVar = "Hello";

boolean boolVar = true;

System.out.println("Integer: " + intVar);

System.out.println("Float: " + floatVar);

System.out.println("Character: " + charVar);

System.out.println("String: " + stringVar);

System.out.println("Boolean: " + boolVar);

}

}

1. **Write a Java program to swap the values of two integer variables.**

public class SwapValues {

public static void main(String[] args) {

int a = 5;

int b = 10;

int temp = a;

a = b;

b = temp;

System.out.println("a: " + a);

System.out.println("b: " + b);

}

}

1. **Write a Java program to swap the values of two integer variables without using a third variable.**

public class SwapWithoutTemp {

public static void main(String[] args) {

int a = 5;

int b = 10;

a = a + b;

b = a - b;

a = a - b;

System.out.println("a: " + a);

System.out.println("b: " + b);

}

}

1. **Write a Java program to calculate the area of a rectangle given its length and width.**

public class RectangleArea {

public static void main(String[] args) {

double length = 5.0;

double width = 3.0;

double area = length \* width;

System.out.println("Area: " + area);

}

}

1. **Write a Java program to find the sum, difference, product, and quotient of two integers.**

public class ArithmeticOperations {

public static void main(String[] args) {

int num1 = 15;

int num2 = 5;

int sum = num1 + num2;

int difference = num1 - num2;

int product = num1 \* num2;

int quotient = num1 / num2;

System.out.println("Sum: " + sum);

System.out.println("Difference: " + difference);

System.out.println("Product: " + product);

System.out.println("Quotient: " + quotient);

}

}

1. **Write a Java program to check if a number is even or odd.**

public class EvenOddTernary {

public static void main(String[] args) {

int number = 7;

String result = (number % 2 == 0) ? "Even" : "Odd";

System.out.println(result);

}

}

1. **Write a Java program to calculate the area of a Triangle given its base and height.**

public class TriangleArea {

public static void main(String[] args) {

// Define the base and height of the triangle

double base = 5.0; // example value

double height = 10.0; // example value

// Calculate the area of the triangle

double area = (base \* height) / 2;

// Display the result

System.out.println("The area of the triangle is: " + area);

}

}

1. **Write a Java program to calculate the area and Perimeter of a Circle given its radius.**

public class CircleCalculations {

public static void main(String[] args) {

// Define the radius of the circle

double radius = 7.0; // example value

// Calculate the area of the circle

double area = Math.PI \* radius \* radius;

// Calculate the perimeter (circumference) of the circle

double perimeter = 2 \* Math.PI \* radius;

// Display the results

System.out.println("The area of the circle is: " + area);

System.out.println("The perimeter (circumference) of the circle is: " + perimeter);

}

}

1. **Write a Java program to find the remainder when one integer is divided by another.**

public class Remainder {

public static void main(String[] args) {

int num1 = 10;

int num2 = 3;

int remainder = num1 % num2;

System.out.println("Remainder: " + remainder);

}

}

1. **Write a Java program to compare two integers and print the larger one.**

public class CompareIntegersTernary {

public static void main(String[] args) {

int num1 = 15;

int num2 = 10;

int largerNumber = (num1 > num2) ? num1 : num2;

System.out.println("Larger number: " + largerNumber);

}

}

1. **Write a Java program to calculate the perimeter of a rectangle given its length and width.**

public class RectanglePerimeter {

public static void main(String[] args) {

double length = 5.0;

double width = 3.0;

double perimeter = 2 \* (length + width);

System.out.println("Perimeter: " + perimeter);

}

}

1. **Write a Java program to calculate the average of three integers.**

public class AverageOfThree {

public static void main(String[] args) {

int num1 = 10;

int num2 = 20;

int num3 = 30;

double average = (num1 + num2 + num3) / 3.0;

System.out.println("Average: " + average);

}

}

1. **Write a Java program to convert a given number of seconds into hours, minutes, and seconds.**

public class TimeConversion {

public static void main(String[] args) {

int totalSeconds = 3661;

int hours = totalSeconds / 3600;

int minutes = (totalSeconds % 3600) / 60;

int seconds = totalSeconds % 60;

System.out.println("Hours: " + hours);

System.out.println("Minutes: " + minutes);

System.out.println("Seconds: " + seconds);

}

}

1. **Write a Java program to calculate the area and circumference of a circle given its radius.**

public class Circle {

public static void main(String[] args) {

double radius = 5.0;

double area = Math.PI \* radius \* radius;

double circumference = 2 \* Math.PI \* radius;

System.out.println("Area: " + area);

System.out.println("Circumference: " + circumference);

}

}

1. **Write a Java program to declare a boolean variable, assign a value to it, and print the value.**

public class BooleanVariable {

public static void main(String[] args) {

boolean isJavaFun = true;

System.out.println("Is Java fun? " + isJavaFun);

}

}

1. **Write a Java program to convert temperatures from Fahrenheit to Celsius.**

public class FahrenheitToCelsius {

public static void main(String[] args) {

double fahrenheit = 98.6;

double celsius = (fahrenheit - 32) \* 5 / 9;

System.out.println("Celsius: " + celsius);

}

}

1. **Write a program to check if a given number is even or odd.**

import java.util.Scanner;

public class EvenOdd {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (number % 2 == 0) {

System.out.println(number + " is even.");

} else {

System.out.println(number + " is odd.");

}

}

}

1. **Write a program to determine if a given number is positive, negative, or zero.**

import java.util.Scanner;

public class PositiveNegative {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (number > 0) {

System.out.println(number + " is positive.");

} else if (number < 0) {

System.out.println(number + " is negative.");

} else {

System.out.println(number + " is zero.");

}

}

}

1. **Write a program that checks if a given character is a vowel or consonant.**

import java.util.Scanner;

public class VowelConsonant {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

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

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||

ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {

System.out.println(ch + " is a vowel.");

} else {

System.out.println(ch + " is a consonant.");

}

}

}

1. **Write a program to find the largest of two numbers.**

import java.util.Scanner;

public class LargestOfTwo {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int num1 = scanner.nextInt();

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

int num2 = scanner.nextInt();

if (num1 > num2) {

System.out.println(num1 + " is larger.");

} else if (num2 > num1) {

System.out.println(num2 + " is larger.");

} else {

System.out.println("Both numbers are equal.");

}

}

}

1. **Write a program to check if a given year is a leap year.**

import java.util.Scanner;

public class LeapYear {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int year = scanner.nextInt();

if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {

System.out.println(year + " is a leap year.");

} else {

System.out.println(year + " is not a leap year.");

}

}

}

1. **Write a program to check if a student passed or failed based on marks (pass mark is 50).**

import java.util.Scanner;

public class PassFail {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the marks: ");

int marks = scanner.nextInt();

if (marks >= 50) {

System.out.println("Pass");

} else {

System.out.println("Fail");

}

}

}

1. **Write a program to check if a number is divisible by 5 and 11.**

import java.util.Scanner;

public class Divisibility {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (number % 5 == 0 && number % 11 == 0) {

System.out.println(number + " is divisible by both 5 and 11.");

} else {

System.out.println(number + " is not divisible by both 5 and 11.");

}

}

}

1. **Write a program to determine the age group of a person (child, teenager, adult, senior) based on their age.**

import java.util.Scanner;

public class AgeGroup {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the age: ");

int age = scanner.nextInt();

if (age >= 0 && age <= 12) {

System.out.println("Child");

} else if (age >= 13 && age <= 19) {

System.out.println("Teenager");

} else if (age >= 20 && age <= 59) {

System.out.println("Adult");

} else if (age >= 60) {

System.out.println("Senior");

} else {

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

}

}

}

1. **Write a simple calculator program that performs addition, subtraction, multiplication, and division based on user input.**

import java.util.Scanner;

public class SimpleCalculator {

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 second number: ");

double num2 = scanner.nextDouble();

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

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

double result;

if (operator == '+') {

result = num1 + num2;

} else if (operator == '-') {

result = num1 - num2;

} else if (operator == '\*') {

result = num1 \* num2;

} else if (operator == '/') {

result = num1 / num2;

} else {

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

return;

}

System.out.println("The result is: " + result);

}

}

1. **Write a program to check if a given character is an uppercase or lowercase letter.**

import java.util.Scanner;

public class UpperLower {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

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

if (ch >= 'A' && ch <= 'Z') {

System.out.println(ch + " is an uppercase letter.");

} else if (ch >= 'a' && ch <= 'z') {

System.out.println(ch + " is a lowercase letter.");

} else {

System.out.println(ch + " is not a letter.");

}

}

}

1. **Write a program to compare three numbers and print the largest one.**

public class NumberComparison {

public static void main(String[] args) {

int a = 10, b = 20, c = 15;

int largest;

if (a >= b && a >= c) {

largest = a;

} else if (b >= a && b >= c) {

largest = b;

} else {

largest = c;

}

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

}

}

1. **Write a program to calculate the grade of a student based on marks using an if-else-if ladder.**

public class GradeCalculation {

public static void main(String[] args) {

int marks = 85;

char grade;

if (marks >= 90) {

grade = 'O';

} else if (marks >= 80) {

grade = 'E';

} else if (marks >= 70) {

grade = 'A';

} else if (marks >= 60) {

grade = 'B';

} else if (marks >= 50) {

grade = 'C';

} else if (marks >= 40) {

grade = 'D';

}else {

grade = 'F';

}

System.out.println("The grade is: " + grade);

}

}

1. **Write a program to print the day of the week based on a number input (1 for Monday, 2 for Tuesday, etc.).**

public class DayOfWeek {

public static void main(String[] args) {

int day = 3;

String dayName;

switch (day) {

case 1: dayName = "Monday"; break;

case 2: dayName = "Tuesday"; break;

case 3: dayName = "Wednesday"; break;

case 4: dayName = "Thursday"; break;

case 5: dayName = "Friday"; break;

case 6: dayName = "Saturday"; break;

case 7: dayName = "Sunday"; break;

default: dayName = "Invalid day"; break;

}

System.out.println("The day is: " + dayName);

}

}

1. **Write a program to find the roots of a quadratic equation.**

public class QuadraticEquationRoots {

public static void main(String[] args) {

double a = 1, b = -3, c = 2;

double determinant = b \* b - 4 \* a \* c;

double root1, root2;

if (determinant > 0) {

root1 = (-b + Math.sqrt(determinant)) / (2 \* a);

root2 = (-b - Math.sqrt(determinant)) / (2 \* a);

System.out.println("Roots are: " + root1 + " and " + root2);

} else if (determinant == 0) {

root1 = root2 = -b / (2 \* a);

System.out.println("Roots are: " + root1 + " and " + root2);

} else {

System.out.println("Roots are imaginary");

}

}

}

1. **Write a program to check if a person is eligible to vote based on their age and nationality.**

public class VotingEligibility {

public static void main(String[] args) {

int age = 20;

String nationality = "Indian";

if (age >= 18 && nationality ="Indian") {

System.out.println("Eligible to vote");

} else {

System.out.println("Not eligible to vote");

}

}

}

1. **Write a program to determine the type of triangle (equilateral, isosceles, scalene) based on side lengths.**

public class TriangleType {

public static void main(String[] args) {

int side1 = 5, side2 = 5, side3 = 5;

if (side1 == side2 && side2 == side3) {

System.out.println("The triangle is Equilateral");

} else if (side1 == side2 || side1 == side3 || side2 == side3) {

System.out.println("The triangle is Isosceles");

} else {

System.out.println("The triangle is Scalene");

}

}

}

1. **Write a program to check if three given sides form a valid triangle.**

import java.util.Scanner;

public class TriangleValidity {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input three sides of the triangle

System.out.print("Enter the length of side 1: ");

int side1 = scanner.nextInt();

System.out.print("Enter the length of side 2: ");

int side2 = scanner.nextInt();

System.out.print("Enter the length of side 3: ");

int side3 = scanner.nextInt();

// Check if the given sides form a valid triangle

if (isValidTriangle(side1, side2, side3)) {

System.out.println("The given sides form a valid triangle.");

} else {

System.out.println("The given sides do not form a valid triangle.");

}

scanner.close();

}

// Method to check if three sides form a valid triangle

public static boolean isValidTriangle(int side1, int side2, int side3) {

// Check the triangle inequality theorem

return (side1 + side2 > side3) && (side1 + side3 > side2) && (side2 + side3 > side1);

}

}

1. **Write a program in java to find the middle number among three given numbers.**

import java.util.Scanner;

public class MiddleNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input three numbers

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

int num1 = scanner.nextInt();

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

int num2 = scanner.nextInt();

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

int num3 = scanner.nextInt();

// Find and print the middle number

int middle = findMiddleNumber(num1, num2, num3);

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

scanner.close();

}

// Method to find the middle number among three given numbers

public static int findMiddleNumber(int num1, int num2, int num3) {

if ((num1 > num2 && num1 < num3) || (num1 > num3 && num1 < num2)) {

return num1;

} else if ((num2 > num1 && num2 < num3) || (num2 > num3 && num2 < num1)) {

return num2;

} else {

return num3;

}

}

}

1. **Write a program in java to check if a number is a perfect square.**

import java.util.Scanner;

public class PerfectSquareCheck {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input number to check for perfect square

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

int number = scanner.nextInt();

// Check if the number is a perfect square

if (isPerfectSquare(number)) {

System.out.println(number + " is a perfect square.");

} else {

System.out.println(number + " is not a perfect square.");

}

scanner.close();

}

// Method to check if a number is a perfect square

public static boolean isPerfectSquare(int number) {

int sqrt = (int) Math.sqrt(number);

return (sqrt \* sqrt == number);

}

}

1. **Write a program to calculate the BMI and categorize it (underweight, normal, overweight, obese).**

import java.util.Scanner;

public class BMI {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter weight in kilograms: ");

double weight = scanner.nextDouble();

System.out.print("Enter height in meters: ");

double height = scanner.nextDouble();

double bmi = weight / (height \* height);

System.out.println("BMI: " + bmi);

if (bmi < 18.5) {

System.out.println("Underweight");

} else if (bmi >= 18.5 && bmi < 24.9) {

System.out.println("Normal weight");

} else if (bmi >= 25 && bmi < 29.9) {

System.out.println("Overweight");

} else {

System.out.println("Obese");

}

scanner.close();

}

}

1. **Check if a Number is an Automorphic Number.**
2. **A**
3. **Print "Hello World" 5 times**

public class HelloWorld {

public static void main(String[] args) {

for (int i = 0; i < 5; i++) {

System.out.println("Hello World");

}

}

}

1. **Print Numbers from 1 to 10**

public class PrintNumbers {

public static void main(String[] args) {

for (int i = 1; i <= 10; i++) {

System.out.println(i);

}

}

}

1. **Print Even Numbers from 1 to 10**

public class PrintEvenNumbers {

public static void main(String[] args) {

for (int i = 1; i <= 10; i++) {

if (i % 2 == 0) {

System.out.println(i);

}

}

}

}

1. **Print Odd Numbers from 1 to 10**

public class PrintOddNumbers {

public static void main(String[] args) {

for (int i = 1; i <= 10; i++) {

if (i % 2 != 0) {

System.out.println(i);

}

}

}

}

1. **Sum of Numbers from 1 to 5**

public class SumNumbers {

public static void main(String[] args) {

int sum = 0;

for (int i = 1; i <= 5; i++) {

sum += i;

}

System.out.println("Sum: " + sum);

}

}

1. **Print Multiplication Table of 2**

public class MultiplicationTable {

public static void main(String[] args) {

int number = 2;

for (int i = 1; i <= 10; i++) {

System.out.println(number + " x " + i + " = " + (number \* i));

}

}

}

1. **Print Characters of a String**

public class PrintCharacters {

public static void main(String[] args) {

String text = "Hello";

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

System.out.println(text.charAt(i));

}

}

}

1. **Reverse Counting from 10 to 1**

public class ReverseCounting {

public static void main(String[] args) {

for (int i = 10; i >= 1; i--) {

System.out.println(i);

}

}

}

1. **Print "Java" if Number is Divisible by 3 (1 to 10)**

public class DivisibleByThree {

public static void main(String[] args) {

for (int i = 1; i <= 10; i++) {

if (i % 3 == 0) {

System.out.println("Java");

} else {

System.out.println(i);

}

}

}

}

1. **Print "0" if Number is Divisible by either 2 or 5 (1 to 10)**

public class DivisibleByTwoOrFive {

public static void main(String[] args) {

for (int i = 1; i <= 10; i++) {

if (i % 2 == 0 || i % 5 == 0) {

System.out.println("0");

} else {

System.out.println(i);

}

}

}

}

1. **Sum of Even Numbers from 1 to 10**

public class SumEvenNumbers {

public static void main(String[] args) {

int sum = 0;

for (int i = 1; i <= 10; i++) {

if (i % 2 == 0) {

sum += i;

}

}

System.out.println("Sum of even numbers: " + sum);

}

}

1. **Factorial of a Number**

public class Factorial {

public static void main(String[] args) {

int number = 5; // Example number

int factorial = 1;

for (int i = 1; i <= number; i++) {

factorial \*= i;

}

System.out.println("Factorial of " + number + " is " + factorial);

}

}

1. **Check Prime Number**

public class PrimeNumber {

public static void main(String[] args) {

int number = 7; // Example number

boolean isPrime = true;

for (int i = 2; i <= number / 2; i++) {

if (number % i == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

System.out.println(number + " is a prime number.");

} else {

System.out.println(number + " is not a prime number.");

}

}

}

1. **Print Fibonacci Series**

public class FibonacciSeries {

public static void main(String[] args) {

int n = 10; // Number of terms

int first = 0, second = 1;

System.out.print(first + " " + second);

for (int i = 2; i < n; i++) {

int next = first + second;

System.out.print(" " + next);

first = second;

second = next;

}

}

}

1. **Count Digits in a Number**

public class CountDigits {

public static void main(String[] args) {

int number = 12345; // Example number

int count = 0;

while (number != 0) {

number /= 10;

count++;

}

System.out.println("Number of digits: " + count);

}

}

1. **Reverse a String**

public class ReverseString {

public static void main(String[] args) {

String text = "Hello";

String reversed = "";

for (int i = text.length() - 1; i >= 0; i--) {

reversed += text.charAt(i);

}

System.out.println("Reversed string: " + reversed);

}

}

1. **Check Palindrome String**

public class PalindromeString {

public static void main(String[] args) {

String text = "madam";

String reversed = "";

for (int i = text.length() - 1; i >= 0; i--) {

reversed += text.charAt(i);

}

if (text.equals(reversed)) {

System.out.println(text + " is a palindrome.");

} else {

System.out.println(text + " is not a palindrome.");

}

}

}

1. **Write a Java program to calculate the sum of the digits of a given number.**

public class SumOfDigits {

public static void main(String[] args) {

int number = 123; // Example number

int sum = 0;

while (number != 0) {

int digit = number % 10; // Get the last digit

sum += digit; // Add the digit to the sum

number /= 10; // Remove the last digit

}

System.out.println("Sum of digits: " + sum);

}

}

1. **Print a right-angled triangle pattern of stars with 5 rows.**

public class RightAngledTriangle {

public static void main(String[] args) {

int rows = 5; // Number of rows for the triangle

for (int i = 1; i <= rows; i++) {

for (int j = 1; j <= i; j++) {

System.out.print("\* ");

}

System.out.println();

}

}

}

1. **Print a mirrored right-angled triangle pattern of stars with 5 rows.**

**public class MirroredRightAngledTriangle {**

**public static void main(String[] args) {**

**int rows = 5; // Number of rows for the triangle**

**for (int i = 1; i <= rows; i++) {**

**for (int j = i; j < rows; j++) {**

**System.out.print(" ");**

**}**

**for (int k = 1; k <= i; k++) {**

**System.out.print("\* ");**

**}**

**System.out.println();**

**}**

**}**

**}**

1. **Print Patterns (Pyramid)**

public class TrianglePattern {

public static void main(String[] args) {

int rows = 5;

for (int i = 1; i <= rows; i++) {

for (int j = 1; j <= i; j++) {

System.out.print("\* ");

}

System.out.println();

}

}

}

1. **Print Patterns (Diamond)**

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

import java.util.Scanner;

public class DiamondPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int rows = scanner.nextInt();

int n = rows / 2 + 1;

// Print the upper part of the diamond

for (int i = 1; i <= n; i++) {

// Print spaces

for (int j = i; j < n; j++) {

System.out.print(" ");

}

// Print stars

for (int j = 1; j <= (2 \* i - 1); j++) {

System.out.print("\*");

}

System.out.println();

}

// Print the lower part of the diamond

for (int i = n - 1; i >= 1; i--) {

// Print spaces

for (int j = n; j > i; j--) {

System.out.print(" ");

}

// Print stars

for (int j = 1; j <= (2 \* i - 1); j++) {

System.out.print("\*");

}

System.out.println();

}

scanner.close();

}

}

1. **Print the first 10 prime numbers.**

public class FirstTenPrimes {

public static void main(String[] args) {

int count = 0;

int number = 2;

System.out.println("First 10 prime numbers:");

while (count < 10) {

if (isPrime(number)) {

System.out.print(number + " ");

count++;

}

number++;

}

}

public static boolean isPrime(int num) {

if (num <= 1) {

return false;

}

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

}

1. **Check whether a number is an Armstrong number.**

public class ArmstrongNumber {

public static void main(String[] args) {

int number = 153;

if (isArmstrong(number)) {

System.out.println(number + " is an Armstrong number.");

} else {

System.out.println(number + " is not an Armstrong number.");

}

}

public static boolean isArmstrong(int num) {

int originalNum = num;

int sum = 0;

int digits = String.valueOf(num).length();

while (num > 0) {

int digit = num % 10;

sum += Math.pow(digit, digits);

num /= 10;

}

return sum == originalNum;

}

}

1. **Generate the first 15 numbers in the Fibonacci sequence.**

public class FibonacciSequence {

public static void main(String[] args) {

int n = 15;

System.out.println("First " + n + " Fibonacci numbers:");

int a = 0, b = 1;

for (int i = 1; i <= n; i++) {

System.out.print(a + " ");

int sum = a + b;

a = b;

b = sum;

}

}

}

1. **Print the sum of squares of the first 10 natural numbers.**

public class SumOfSquares {

public static void main(String[] args) {

int n = 10;

int sum = 0;

for (int i = 1; i <= n; i++) {

sum += i \* i;

}

System.out.println("Sum of squares of first " + n + " natural numbers is: " + sum);

}

}

1. **print the GCD of two numbers.**

public class GCD {

public static void main(String[] args) {

int a = 24;

int b = 18;

System.out.println("GCD of " + a + " and " + b + " is: " + findGCD(a, b));

}

public static int findGCD(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

}

1. **Print the reverse of a given number.**

public class ReverseNumber {

public static void main(String[] args) {

int number = 12345;

int reversed = reverse(number);

System.out.println("Reverse of " + number + " is: " + reversed);

}

public static int reverse(int num) {

int reversed = 0;

while (num != 0) {

int digit = num % 10;

reversed = reversed \* 10 + digit;

num /= 10;

}

return reversed;

}

}

1. **Print the sum of the first 100 natural numbers.**

public class SumOfNaturalNumbers {

public static void main(String[] args) {

int n = 100;

int sum = n \* (n + 1) / 2;

System.out.println("Sum of first " + n + " natural numbers is: " + sum);

}

}

1. **Print the first 10 terms of the arithmetic sequence with a given initial term and common difference.**

public class ArithmeticSequence {

public static void main(String[] args) {

int initialTerm = 2;

int commonDifference = 3;

int terms = 10;

System.out.println("First " + terms + " terms of the arithmetic sequence:");

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

System.out.print(initialTerm + i \* commonDifference + " ");

}

}

}

1. **Print a hollow square pattern of stars with a given side length.**

public class HollowSquarePattern {

public static void main(String[] args) {

int sideLength = 5;

for (int i = 1; i <= sideLength; i++) {

for (int j = 1; j <= sideLength; j++) {

if (i == 1 || i == sideLength || j == 1 || j == sideLength) {

System.out.print("\* ");

} else {

System.out.print(" ");

}

}

System.out.println();

}

}

}

1. **Check whether a given number is a perfect number.**

public class PerfectNumber {

public static void main(String[] args) {

int number = 28;

if (isPerfectNumber(number)) {

System.out.println(number + " is a perfect number.");

} else {

System.out.println(number + " is not a perfect number.");

}

}

public static boolean isPerfectNumber(int num) {

if (num <= 1) {

return false;

}

int sum = 1;

for (int i = 2; i \* i <= num; i++) {

if (num % i == 0) {

sum += i;

if (i != num / i) {

sum += num / i;

}

}

}

return sum == num;

}

}

1. **Calculate the LCM of two numbers.**

public class LCM {

public static void main(String[] args) {

int a = 12;

int b = 18;

System.out.println("LCM of " + a + " and " + b + " is: " + findLCM(a, b));

}

public static int findLCM(int a, int b) {

return (a \* b) / findGCD(a, b);

}

public static int findGCD(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

}

1. **Print a right-angled triangle pattern of numbers.**

public class NumberTriangle {

public static void main(String[] args) {

int rows = 5;

for (int i = 1; i <= rows; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(j + " ");

}

System.out.println();

}

}

}

1. **Print a pattern of alternating stars and dashes (e.g., "- \*").**

public class AlternatingPattern {

public static void main(String[] args) {

int n = 5;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

if (j % 2 == 0) {

System.out.print("- ");

} else {

System.out.print("\* ");

}

}

System.out.println();

}

}

}

1. **Print Pascal's Triangle**

public class PascalsTriangle {

public static void main(String[] args) {

int rows = 5;

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

            // Print spaces for formatting

            for (int j = 0; j < rows - i; j++) {

                System.out.print(" ");

            }

            int number = 1;

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

                System.out.print(number + " ");

                number = number \* (i - j) / (j + 1);

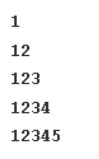
            }

            System.out.println();

        }

}

}

1. ****

public class PK {

    public static void main(String[] args) {

        int rows = 5; // Number of rows in the pattern

        for (int i = 1; i <= rows; i++) {

            for (int j = 1; j <= i; j++) {

                System.out.print(j);

            }

            System.out.println();

        }

    }

}

1. ****

public class PK {

    public static void main(String[] args) {

        int rows = 5; // Number of rows in the pattern

        for (int i = 1; i <= rows; i++) {

            for (int j = 1; j <= i; j++) {

                System.out.print(i);

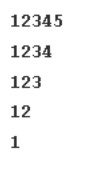
            }

            System.out.println();

        }

    }

}

1. ****

public class PK {

    public static void main(String[] args) {

        int n = 5; // Number of rows in the pattern

        for (int i = n; i >= 1; i--) {

            for (int j = 1; j <= i; j++) {

                System.out.print(j);

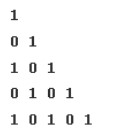
            }

            System.out.println();

        }

    }

}

1. ****

public class PK {

    public static void main(String[] args) {

        int n = 5; // Number of rows in the pattern

        for (int i = 1; i <= n; i++) {

            for (int j = 1; j <= i; j++) {

                // Print 1 if the sum of i and j is even, else print 0

                if ((i + j) % 2 == 0) {

                    System.out.print("1 ");

                } else {

                    System.out.print("0 ");

                }

            }

            System.out.println();

        }

    }

}

***FUNCTION***

1. **Calculate Sum of Two Numbers**
   * Write a function that takes two integers as arguments and returns their sum.

public class Main {

public static int sum(int a, int b) {

return a + b;

}

public static void main(String[] args) {

System.out.println(sum(5, 3)); // Output: 8

}

}

1. **Check Even or Odd**
   * Write a function that takes an integer as an argument and returns "Even" if the number is even and "Odd" if the number is odd.

public class Main {

public static String checkEvenOdd(int num) {

return (num % 2 == 0) ? "Even" : "Odd";

}

public static void main(String[] args) {

System.out.println(checkEvenOdd(7)); // Output: Odd

}

}

1. **Find Maximum of Two Numbers**
   * Write a function that takes two integers as arguments and returns the larger of the two.

public class Main {

public static int max(int a, int b) {

return (a > b) ? a : b;

}

public static void main(String[] args) {

System.out.println(max(5, 3)); // Output: 5

}

}

1. **Calculate Factorial**
   * Write a function that takes an integer as an argument and returns its factorial.

public class Main {

public static int factorial(int n) {

if (n == 0) return 1;

return n \* factorial(n - 1);

}

public static void main(String[] args) {

System.out.println(factorial(5)); // Output: 120

}

}

1. **Check Prime Number**
   * Write a function that takes an integer as an argument and returns true if the number is prime, otherwise false.

public class Main {

public static boolean isPrime(int n) {

if (n <= 1) return false;

for (int i = 2; i <= Math.sqrt(n); i++) {

if (n % i == 0) return false;

}

return true;

}

public static void main(String[] args) {

System.out.println(isPrime(11)); // Output: true

}

}

1. **Calculate Power of a Number**
   * Write a function that takes two integers, base and exponent, and returns the result of base raised to the power of exponent.

public class Main {

public static int power(int base, int exponent) {

return (int) Math.pow(base, exponent);

}

public static void main(String[] args) {

System.out.println(power(2, 3)); // Output: 8

}

}

1. **Find the Length of a String**
   * Write a function that takes a string as an argument and returns its length.

public class Main {

public static int length(String str) {

return str.length();

}

public static void main(String[] args) {

System.out.println(length("Hello")); // Output: 5

}

}

1. **Convert Celsius to Fahrenheit**
   * Write a function that takes a temperature in Celsius and returns the equivalent temperature in Fahrenheit.

public class Main {

public static double celsiusToFahrenheit(double celsius) {

return (celsius \* 9/5) + 32;

}

public static void main(String[] args) {

System.out.println(celsiusToFahrenheit(0)); // Output: 32.0

}

}

1. **Calculate Area of a Rectangle**
   * Write a function that takes the length and width of a rectangle and returns its area.

public class Main {

public static int area(int length, int width) {

return length \* width;

}

public static void main(String[] args) {

System.out.println(area(5, 3)); // Output: 15

}

}

1. **Check Palindrome**
   * Write a function that takes a string as an argument and returns true if the string is a palindrome, otherwise false.

public class Main {

public static boolean isPalindrome(String str) {

int n = str.length();

for (int i = 0; i < n / 2; i++) {

if (str.charAt(i) != str.charAt(n - 1 - i)) return false;

}

return true;

}

public static void main(String[] args) {

System.out.println(isPalindrome("radar")); // Output: true

}

}

**Easy Level**

1. **Calculate GCD**
   * Write a function that takes two integers as arguments and returns their greatest common divisor (GCD).

public class Main {

public static int gcd(int a, int b) {

if (b == 0) return a;

return gcd(b, a % b);

}

public static void main(String[] args) {

System.out.println(gcd(54, 24)); // Output: 6

}

}

1. **Reverse a String**
   * Write a function that takes a string as an argument and returns the reversed string.

public class Main {

public static String reverse(String str) {

return new StringBuilder(str).reverse().toString();

}

public static void main(String[] args) {

System.out.println(reverse("Hello")); // Output: olleH

}

}

1. **Check Armstrong Number**
   * Write a function that takes an integer as an argument and returns true if the number is an Armstrong number, otherwise false.

public class Main {

public static boolean isArmstrong(int num) {

int original = num, sum = 0, digits = String.valueOf(num).length();

while (num != 0) {

int digit = num % 10;

sum += Math.pow(digit, digits);

num /= 10;

}

return sum == original;

}

public static void main(String[] args) {

System.out.println(isArmstrong(153)); // Output: true

}

}

1. **Find Minimum of Three Numbers**
   * Write a function that takes three integers as arguments and returns the smallest of the three.

public class Main {

public static int min(int a, int b, int c) {

return Math.min(a, Math.min(b, c));

}

public static void main(String[] args) {

System.out.println(min(5, 3, 7)); // Output: 3

}

}

1. **Calculate Sum of Digits**
   * Write a function that takes an integer as an argument and returns the sum of its digits.

public class Main {

public static int sumOfDigits(int num) {

int sum = 0;

while (num != 0) {

sum += num % 10;

num /= 10;

}

return sum;

}

public static void main(String[] args) {

System.out.println(sumOfDigits(1234)); // Output: 10

}

}

1. **Check Perfect Number**
   * Write a function that takes an integer as an argument and returns true if the number is a perfect number, otherwise false.

public class Main {

public static boolean isPerfect(int num) {

int sum = 0;

for (int i = 1; i < num; i++) {

if (num % i == 0) sum += i;

}

return sum == num;

}

public static void main(String[] args) {

System.out.println(isPerfect(28)); // Output: true

}

}

1. **Count Vowels in a String**
   * Write a function that takes a string as an argument and returns the number of vowels in the string.

public class Main {

public static int countVowels(String str) {

int count = 0;

for (char c : str.toLowerCase().toCharArray()) {

if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {

count++;

}

}

return count;

}

public static void main(String[] args) {

System.out.println(countVowels("Hello World")); // Output: 3

}

}

1. **Find Fibonacci Series**
   * Write a function that takes an integer n as an argument and returns the first n Fibonacci numbers.

public class Main {

public static int[] fibonacci(int n) {

int[] fib = new int[n];

fib[0] = 0;

if (n > 1) fib[1] = 1;

for (int i = 2; i < n; i++) {

fib[i] = fib[i - 1] + fib[i - 2];

}

return fib;

}

public static void main(String[] args) {

int[] result = fibonacci(10);

for (int num : result) {

System.out.print(num + " ");

}

// Output: 0 1 1 2 3 5 8 13 21 34

}

}

1. **Convert Binary to Decimal**
   * Write a function that takes a binary string as an argument and returns its decimal equivalent.

public class Main {

public static int binaryToDecimal(String binary) {

return Integer.parseInt(binary, 2);

}

public static void main(String[] args) {

System.out.println(binaryToDecimal("1010")); // Output: 10

}

}

1. **Check Anagram**
   * Write a function that takes two strings as arguments and returns true if they are anagrams, otherwise false.

import java.util.Arrays;

public class Main {

public static boolean isAnagram(String str1, String str2) {

char[] arr1 = str1.toCharArray();

char[] arr2 = str2.toCharArray();

Arrays.sort(arr1);

Arrays.sort(arr2);

return Arrays.equals(arr1, arr2);

}

public static void main(String[] args) {

System.out.println(isAnagram("listen", "silent")); // Output: true

}

}

**Medium Level**

1. **Find LCM**
   * Write a function that takes two integers as arguments and returns their least common multiple (LCM).

public class Main {

public static int gcd(int a, int b) {

if (b == 0) return a;

return gcd(b, a % b);

}

public static int lcm(int a, int b) {

return (a \* b) / gcd(a, b);

}

public static void main(String[] args) {

System.out.println(lcm(12, 18)); // Output: 36

}

}

1. **Sort an Array**
   * Write a function that takes an array of integers as an argument and returns the sorted array.

import java.util.Arrays;

public class Main {

public static int[] sortArray(int[] arr) {

Arrays.sort(arr);

return arr;

}

public static void main(String[] args) {

int[] array = {5, 3, 8, 1, 2};

array = sortArray(array);

System.out.println(Arrays.toString(array)); // Output: [1, 2, 3, 5, 8]

}

}

1. **Find Nth Prime Number**
   * Write a function that takes an integer n as an argument and returns the nth prime number.

public class Main {

public static boolean isPrime(int n) {

if (n <= 1) return false;

for (int i = 2; i <= Math.sqrt(n); i++) {

if (n % i == 0) return false;

}

return true;

}

public static int nthPrime(int n) {

int count = 0;

int num = 1;

while (count < n) {

num++;

if (isPrime(num)) {

count++;

}

}

return num;

}

public static void main(String[] args) {

System.out.println(nthPrime(10)); // Output: 29

}

}

1. **Remove Duplicates from Array**
   * Write a function that takes an array of integers as an argument and returns a new array with duplicates removed.

import java.util.Arrays;

import java.util.HashSet;

public class Main {

public static int[] removeDuplicates(int[] arr) {

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

for (int num : arr) {

set.add(num);

}

int[] result = new int[set.size()];

int i = 0;

for (int num : set) {

result[i++] = num;

}

return result;

}

public static void main(String[] args) {

int[] array = {1, 2, 3, 2, 4, 5, 1};

array = removeDuplicates(array);

System.out.println(Arrays.toString(array)); // Output: [1, 2, 3, 4, 5]

}

}

1. **Check Happy Number**
   * Write a function that takes an integer as an argument and returns true if the number is a happy number, otherwise false.

import java.util.HashSet;

public class Main {

public static boolean isHappy(int num) {

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

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

seen.add(num);

num = sumOfSquares(num);

}

return num == 1;

}

private static int sumOfSquares(int num) {

int sum = 0;

while (num != 0) {

int digit = num % 10;

sum += digit \* digit;

num /= 10;

}

return sum;

}

public static void main(String[] args) {

System.out.println(isHappy(19)); // Output: true

}

}

1. **Calculate Compound Interest**
   * Write a function that takes principal, rate, and time as arguments and returns the compound interest.

public class Main {

public static double compoundInterest(double principal, double rate, int time) {

return principal \* Math.pow((1 + rate / 100), time) - principal;

}

public static void main(String[] args) {

System.out.println(compoundInterest(1000, 5, 2)); // Output: 102.5

}

}

1. **Find Longest Word in a Sentence**
   * Write a function that takes a sentence as an argument and returns the longest word in the sentence.

public class Main {

public static String longestWord(String sentence) {

String[] words = sentence.split("\\s+");

String longest = "";

for (String word : words) {

if (word.length() > longest.length()) {

longest = word;

}

}

return longest;

}

public static void main(String[] args) {

System.out.println(longestWord("The quick brown fox jumps over the lazy dog")); // Output: jumps

}

}

1. **Generate Pascal's Triangle**
   * Write a function that takes an integer n as an argument and returns the first n rows of Pascal's triangle.

public class Main {

public static int[][] pascalTriangle(int n) {

int[][] triangle = new int[n][];

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

triangle[i] = new int[i + 1];

triangle[i][0] = triangle[i][i] = 1;

for (int j = 1; j < i; j++) {

triangle[i][j] = triangle[i - 1][j - 1] + triangle[i - 1][j];

}

}

return triangle;

}

public static void main(String[] args) {

int[][] result = pascalTriangle(5);

for (int[] row : result) {

for (int num : row) {

System.out.print(num + " ");

}

System.out.println();

}

// Output:

// 1

// 1 1

// 1 1 1

// 1 2 1

// 1 3 3 1

// 1 4 6 4 1

}

}

1. **Find Intersection of Two Arrays**
   * Write a function that takes two arrays of integers as arguments and returns an array containing their intersection.

import java.util.Arrays;

import java.util.HashSet;

public class Main {

public static int[] intersection(int[] arr1, int[] arr2) {

HashSet<Integer> set1 = new HashSet<>();

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

for (int num : arr1) {

set1.add(num);

}

for (int num : arr2) {

if (set1.contains(num)) {

intersectionSet.add(num);

}

}

int[] result = new int[intersectionSet.size()];

int i = 0;

for (int num : intersectionSet) {

result[i++] = num;

}

return result;

}

public static void main(String[] args) {

int[] array1 = {1, 2, 3, 4, 5};

int[] array2 = {3, 4, 5, 6, 7};

int[] result = intersection(array1, array2);

System.out.println(Arrays.toString(result)); // Output: [3, 4, 5]

}

}

1. **Find Missing Number in Array**
   * Write a function that takes an array of n-1 integers from 1 to n and returns the missing number.

public class Main {

public static int findMissingNumber(int[] arr, int n) {

int total = n \* (n + 1) / 2;

int sum = 0;

for (int num : arr) {

sum += num;

}

return total - sum;

}

public static void main(String[] args) {

int[] array = {1, 2, 4, 5, 6};

int missing = findMissingNumber(array, 6);

System.out.println(missing); // Output: 3

}

}

***ARRAYS***

1. **Sum of Array Elements**

public class SumArray {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5};

int sum = 0;

for (int number : numbers) {

sum += number;

}

System.out.println("Sum of array elements: " + sum);

}

}

1. **Average of Array Elements**

public class AverageArray {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5};

int sum = 0;

for (int number : numbers) {

sum += number;

}

double average = (double) sum / numbers.length;

System.out.println("Average of array elements: " + average);

}

}

1. **Minimum Element in Array**

public class MinElement {

public static void main(String[] args) {

int[] numbers = {5, 2, 8, 1, 9};

int min = numbers[0];

for (int number : numbers) {

if (number < min) {

min = number;

}

}

System.out.println("Minimum element: " + min);

}

}

1. **Maximum Element in Array**

public class MaxElement {

public static void main(String[] args) {

int[] numbers = {5, 2, 8, 1, 9};

int max = numbers[0];

for (int number : numbers) {

if (number > max) {

max = number;

}

}

System.out.println("Maximum element: " + max);

}

}

1. **Print Array Elements**

public class PrintArray {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5};

for (int number : numbers) {

System.out.print(number + " ");

}

}

}

1. **Reverse an Array**

public class ReverseArray {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5};

for (int i = numbers.length - 1; i >= 0; i--) {

System.out.print(numbers[i] + " ");

}

}

}

1. **Count Even Numbers in Array**

public class CountEven {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5, 6};

int count = 0;

for (int number : numbers) {

if (number % 2 == 0) {

count++;

}

}

System.out.println("Count of even numbers: " + count);

}

}

1. **Count Odd Numbers in Array**

public class CountOdd {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5, 6};

int count = 0;

for (int number : numbers) {

if (number % 2 != 0) {

count++;

}

}

System.out.println("Count of odd numbers: " + count);

}

}

1. **Check if Array Contains a Value**

public class ContainsValue {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5};

int value = 3;

boolean found = false;

for (int number : numbers) {

if (number == value) {

found = true;

break;

}

}

if (found) {

System.out.println("Array contains the value " + value);

} else {

System.out.println("Array does not contain the value " + value);

}

}

}

1. **Find Index of an Element in Array**

public class FindIndex {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5};

int value = 3;

int index = -1;

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

if (numbers[i] == value) {

index = i;

break;

}

}

if (index != -1) {

System.out.println("Index of " + value + " is: " + index);

} else {

System.out.println("Value not found in array");

}

}

}

1. **Sort an Array**

import java.util.Arrays;

public class SortArray {

public static void main(String[] args) {

int[] numbers = {5, 2, 8, 1, 9};

Arrays.sort(numbers);

System.out.println("Sorted array: " + Arrays.toString(numbers));

}

}

1. **Remove Duplicates from Array**

import java.util.Arrays;

public class RemoveDuplicates {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 2, 4, 5, 1};

int[] temp = new int[numbers.length];

int j = 0;

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

boolean isDuplicate = false;

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

if (numbers[i] == temp[k]) {

isDuplicate = true;

break;

}

}

if (!isDuplicate) {

temp[j++] = numbers[i];

}

}

int[] result = Arrays.copyOf(temp, j);

System.out.println("Array without duplicates: " + Arrays.toString(result));

}

}

1. **Merge Two Arrays**

import java.util.Arrays;

public class MergeArrays {

public static void main(String[] args) {

int[] array1 = {1, 2, 3};

int[] array2 = {4, 5, 6};

int[] mergedArray = new int[array1.length + array2.length];

System.arraycopy(array1, 0, mergedArray, 0, array1.length);

System.arraycopy(array2, 0, mergedArray, array1.length, array2.length);

System.out.println("Merged array: " + Arrays.toString(mergedArray));

}

}

1. **Find Second Largest Element in Array**

public class SecondLargest {

public static void main(String[] args) {

int[] numbers = {5, 2, 8, 1, 9};

int first = Integer.MIN\_VALUE;

int second = Integer.MIN\_VALUE;

for (int number : numbers) {

if (number > first) {

second = first;

first = number;

} else if (number > second && number != first) {

second = number;

}

}

System.out.println("Second largest element: " + second);

}

}

1. **Find Second Smallest Element in Array**

public class SecondSmallest {

public static void main(String[] args) {

int[] numbers = {5, 2, 8, 1, 9};

int first = Integer.MAX\_VALUE;

int second = Integer.MAX\_VALUE;

for (int number : numbers) {

if (number < first) {

second = first;

first = number;

} else if (number < second && number != first) {

second = number;

}

}

System.out.println("Second smallest element: " + second);

}

}

1. **Find Common Elements in Two Arrays**

import java.util.Arrays;

public class CommonElements {

public static void main(String[] args) {

int[] array1 = {1, 2, 3, 4, 5};

int[] array2 = {4, 5, 6, 7, 8};

int[] temp = new int[Math.min(array1.length, array2.length)];

int k = 0;

for (int i = 0; i < array1.length; i++) {

for (int j = 0; j < array2.length; j++) {

if (array1[i] == array2[j]) {

boolean isDuplicate = false;

for (int l = 0; l < k; l++) {

if (temp[l] == array1[i]) {

isDuplicate = true;

break;

}

}

if (!isDuplicate) {

temp[k++] = array1[i];

}

}

}

}

int[] result = Arrays.copyOf(temp, k);

System.out.println("Common elements: " + Arrays.toString(result));

}

}

1. **Rotate Array by One Position**

public class RotateArray {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5};

int last = numbers[numbers.length - 1];

for (int i = numbers.length - 1; i > 0; i--) {

numbers[i] = numbers[i - 1];

}

numbers[0] = last;

System.out.println("Array after rotation: " + Arrays.toString(numbers));

}

}

1. **Split Array into Two Parts**

import java.util.Arrays;

public class SplitArray {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5, 6};

int mid = numbers.length / 2;

int[] firstHalf = Arrays.copyOfRange(numbers, 0, mid);

int[] secondHalf = Arrays.copyOfRange(numbers, mid, numbers.length);

System.out.println("First half: " + Arrays.toString(firstHalf));

System.out.println("Second half: " + Arrays.toString(secondHalf));

}

}

1. **Find Missing Number in Array**

public class MissingNumber {

public static void main(String[] args) {

int[] numbers = {1, 2, 4, 5, 6};

int n = numbers.length + 1;

int sumOfN = n \* (n + 1) / 2;

int sumOfArray = 0;

for (int number : numbers) {

sumOfArray += number;

}

int missingNumber = sumOfN - sumOfArray;

System.out.println("Missing number: " + missingNumber);

}

}

1. **Find Duplicates in Array**

import java.util.Arrays;

public class FindDuplicates {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 2, 4, 5, 1};

int[] duplicates = new int[numbers.length];

int k = 0;

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

for (int j = i + 1; j < numbers.length; j++) {

if (numbers[i] == numbers[j]) {

boolean isAlreadyInDuplicates = false;

for (int l = 0; l < k; l++) {

if (duplicates[l] == numbers[i]) {

isAlreadyInDuplicates = true;

break;

}

}

if (!isAlreadyInDuplicates) {

duplicates[k++] = numbers[i];

}

}

}

}

int[] result = Arrays.copyOf(duplicates, k);

System.out.println("Duplicates: " + Arrays.toString(result));

}

}

1. **Write a program to find the Kth largest element in an array.**

**Input: int[] arr = {3, 2, 1, 5, 6, 4}; int k = 2;**

**Output: 5**

import java.util.Arrays;

public class KthLargestElement {

public static void main(String[] args) {

int[] arr = {3, 2, 1, 5, 6, 4};

int k = 2;

System.out.println(findKthLargest(arr, k));

}

public static int findKthLargest(int[] nums, int k) {

Arrays.sort(nums);

return nums[nums.length - k];

}

}

1. **Rotate the elements of an array to the right by K positions.**

**Input: int[] arr = {1, 2, 3, 4, 5, 6, 7}; int k = 3;**

**Output: 5, 6, 7, 1, 2, 3, 4**

import java.util.Arrays;

public class RotateArray {

public static void main(String[] args) {

int[] arr = {1, 2, 3, 4, 5, 6, 7};

int k = 3;

rotate(arr, k);

System.out.println(Arrays.toString(arr));

}

public static void rotate(int[] nums, int k) {

k %= nums.length;

reverse(nums, 0, nums.length - 1);

reverse(nums, 0, k - 1);

reverse(nums, k, nums.length - 1);

}

private static void reverse(int[] nums, int start, int end) {

while (start < end) {

int temp = nums[start];

nums[start] = nums[end];

nums[end] = temp;

start++;

end--;

}

}

}

1. **Find a continuous subarray which adds up to a given number.**

**Input: int[] arr = {1, 2, 3, 7, 5}; int sum = 12;**

**Output: 2, 3, 7**

import java.util.Arrays;

public class SubarraySum {

public static void main(String[] args) {

int[] arr = {1, 2, 3, 7, 5};

int sum = 12;

System.out.println(Arrays.toString(findSubarrayWithSum(arr, sum)));

}

public static int[] findSubarrayWithSum(int[] arr, int sum) {

int currSum = 0;

int start = 0;

for (int end = 0; end < arr.length; end++) {

currSum += arr[end];

while (currSum > sum && start <= end) {

currSum -= arr[start];

start++;

}

if (currSum == sum) {

return Arrays.copyOfRange(arr, start, end + 1);

}

}

return new int[]{};

}

}

1. **Find the majority element in an array.**

**Input: int[] arr = {3, 2, 3};**

**Output: 3**

public class MajorityElement {

public static void main(String[] args) {

int[] arr = {3, 2, 3};

System.out.println(findMajorityElement(arr));

}

public static int findMajorityElement(int[] nums) {

int candidate = findCandidate(nums);

return validateCandidate(nums, candidate) ? candidate : -1;

}

private static int findCandidate(int[] nums) {

int count = 0, candidate = 0;

for (int num : nums) {

if (count == 0) {

candidate = num;

}

count += (num == candidate) ? 1 : -1;

}

return candidate;

}

private static boolean validateCandidate(int[] nums, int candidate) {

int count = 0;

for (int num : nums) {

if (num == candidate) {

count++;

}

}

return count > nums.length / 2;

}

}

1. **Given an array of n elements, with elements ranging from 1 to n, find the missing and repeating number.**

**Input: int[] arr = {4, 3, 6, 2, 1, 1};**

**Output: Missing: 5, Repeating: 1**

import java.util.Arrays;

public class MissingAndRepeating {

public static void main(String[] args) {

int[] arr = {4, 3, 6, 2, 1, 1};

int[] result = findMissingAndRepeating(arr);

System.out.println("Missing: " + result[0] + ", Repeating: " + result[1]);

}

public static int[] findMissingAndRepeating(int[] nums) {

int[] result = new int[2];

int[] count = new int[nums.length + 1];

for (int num : nums) {

count[num]++;

}

for (int i = 1; i < count.length; i++) {

if (count[i] == 0) {

result[0] = i;

} else if (count[i] == 2) {

result[1] = i;

}

}

return result;

}

}

1. **Add One  
   Input: digits = [1,2,3] Output: [1,2,4] Explanation: The array represents the integer 123. Incrementing by one gives 123 + 1 = 124. Thus, the result should be [1,2,4].**

import java.util.Arrays;

public class AddOne {

public static void main(String[] args) {

int[] digits = {1, 2, 3};

System.out.println(Arrays.toString(addOne(digits)));

}

public static int[] addOne(int[] digits) {

for (int i = digits.length - 1; i >= 0; i--) {

if (digits[i] < 9) {

digits[i]++;

return digits;

}

digits[i] = 0;

}

int[] newNumber = new int[digits.length + 1];

newNumber[0] = 1;

return newNumber;

}

}

1. **Single Number  
   Input: nums = [4,1,2,1,2] Output: 4**

public class SingleNumber {

public static void main(String[] args) {

int[] nums = {4, 1, 2, 1, 2};

System.out.println(findSingleNumber(nums));

}

public static int findSingleNumber(int[] nums) {

int single = 0;

for (int num : nums) {

single ^= num;

}

return single;

}

}

1. **Move Zeros  
   Input: nums = [0,1,0,3,12] Output: [1,3,12,0,0]**

import java.util.Arrays;

public class MoveZeros {

public static void main(String[] args) {

int[] nums = {0, 1, 0, 3, 12};

moveZeroes(nums);

System.out.println(Arrays.toString(nums));

}

public static void moveZeroes(int[] nums) {

int insertPos = 0;

for (int num : nums) {

if (num != 0) {

nums[insertPos++] = num;

}

}

while (insertPos < nums.length) {

nums[insertPos++] = 0;

}

}

}

1. **Maximum Consecutive Ones  
   Input: nums = [1,0,1,1,0,1] Output: 2**

public class MaxConsecutiveOnes {

public static void main(String[] args) {

int[] nums = {1, 0, 1, 1, 0, 1};

System.out.println(findMaxConsecutiveOnes(nums));

}

public static int findMaxConsecutiveOnes(int[] nums) {

int maxCount = 0;

int count = 0;

for (int num : nums) {

if (num == 1) {

count++;

maxCount = Math.max(maxCount, count);

} else {

count = 0;

}

}

return maxCount;

}

}

1. **Reshape the Matrix  
   Input: mat = [[1,2],[3,4]], r = 1, c = 4 Output: [[1,2,3,4]]**

import java.util.Arrays;

public class ReshapeMatrix {

public static void main(String[] args) {

int[][] mat = {{1, 2}, {3, 4}};

int r = 1, c = 4;

int[][] result = reshapeMatrix(mat, r, c);

for (int[] row : result) {

System.out.println(Arrays.toString(row));

}

}

public static int[][] reshapeMatrix(int[][] mat, int r, int c) {

int m = mat.length;

int n = mat[0].length;

if (m \* n != r \* c) {

return mat;

}

int[][] reshaped = new int[r][c];

int row = 0, col = 0;

for (int[] oldRow : mat) {

for (int num : oldRow) {

reshaped[row][col] = num;

col++;

if (col == c) {

col = 0;

row++;

}

}

}

return reshaped;

}

}

***Strings in JAVA***

**Very Easy Questions**

* **Write a program to concatenate two given strings.**

public class Main {

public static void main(String[] args) {

String str1 = "Hello, ";

String str2 = "World!";

String result = str1.concat(str2);

System.out.println(result);

}

}

* **Write a program to find the length of a given string.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

int length = str.length();

System.out.println("Length: " + length);

}

}

* **Write a program to find the character at a specific index in a string.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

char ch = str.charAt(7);

System.out.println("Character at index 7: " + ch);

}

}

* **Write a program to extract a substring from a given string.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

String substr = str.substring(7, 12);

System.out.println("Substring: " + substr);

}

}

* **Write a program to compare two strings for equality.**

public class Main {

public static void main(String[] args) {

String str1 = "Hello";

String str2 = "hello";

boolean isEqual = str1.equals(str2);

System.out.println("Strings are equal: " + isEqual);

}

}

* **Write a program to convert a string to uppercase.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

String upperStr = str.toUpperCase();

System.out.println("Uppercase: " + upperStr);

}

}

* **Write a program to convert a string to lowercase.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

String lowerStr = str.toLowerCase();

System.out.println("Lowercase: " + lowerStr);

}

}

* **Write a program to trim whitespace from the beginning and end of a string.**

public class Main {

public static void main(String[] args) {

String str = " Hello, World! ";

String trimmedStr = str.trim();

System.out.println("Trimmed: " + trimmedStr);

}

}

* **Write a program to replace a specific character in a string with another character.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

String replacedStr = str.replace('o', 'a');

System.out.println("Replaced: " + replacedStr);

}

}

* **Write a program to check if a string contains a specific substring.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

boolean contains = str.contains("World");

System.out.println("Contains 'World': " + contains);

}

}

* **Write a program to reverse a given string.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

String reversedStr = new StringBuilder(str).reverse().toString();

System.out.println("Reversed: " + reversedStr);

}

}

* **Write a program to check if a string is empty.**

public class Main {

public static void main(String[] args) {

String str = "";

boolean isEmpty = str.isEmpty();

System.out.println("Is empty: " + isEmpty);

}

}

* **Write a program to convert a string to a character array.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

char[] charArray = str.toCharArray();

System.out.println("Char Array: " + java.util.Arrays.toString(charArray));

}

}

* **Write a program to join an array of strings into a single string with spaces between them.**

public class Main {

public static void main(String[] args) {

String[] words = {"Hello", "World"};

String joinedStr = String.join(" ", words);

System.out.println("Joined: " + joinedStr);

}

}

* **Write a program to check if a string starts with a specific prefix.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

boolean startsWith = str.startsWith("Hello");

System.out.println("Starts with 'Hello': " + startsWith);

}

}

* **Write a program to check if a string ends with a specific suffix.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

boolean endsWith = str.endsWith("World!");

System.out.println("Ends with 'World!': " + endsWith);

}

}

* **Write a program to find the index of the first occurrence of a character in a string.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

int index = str.indexOf('o');

System.out.println("Index of 'o': " + index);

}

}

* **Write a program to find the index of the last occurrence of a character in a string.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

int lastIndex = str.lastIndexOf('o');

System.out.println("Last index of 'o': " + lastIndex);

}

}

* **Write a program to concatenate two strings using StringBuilder.**

public class Main {

public static void main(String[] args) {

StringBuilder sb = new StringBuilder("Hello, ");

sb.append("World!");

System.out.println("Concatenated: " + sb.toString());

}

}

* **Write a program to insert a substring into a string using StringBuffer.**

public class Main {

public static void main(String[] args) {

StringBuffer sb = new StringBuffer("Hello World!");

sb.insert(5, ", Beautiful");

System.out.println("Inserted: " + sb.toString());

}

}

**Easy Questions**

* **Write a program to count the number of vowels in a given string.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

int count = 0;

for (char ch : str.toCharArray()) {

if ("AEIOUaeiou".indexOf(ch) != -1) {

count++;

}

}

System.out.println("Vowel count: " + count);

}

}

* **Write a program to count the number of words in a given string.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

String[] words = str.split("\\s+");

System.out.println("Word count: " + words.length);

}

}

* **Write a program to remove all vowels from a given string.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

String noVowels = str.replaceAll("[AEIOUaeiou]", "");

System.out.println("No vowels: " + noVowels);

}

}

* **Write a program to check if a given string is a palindrome.**

public class Main {

public static void main(String[] args) {

String str = "madam";

String reversedStr = new StringBuilder(str).reverse().toString();

boolean isPalindrome = str.equals(reversedStr);

System.out.println("Is palindrome: " + isPalindrome);

}

}

* **Write a program to check if two given strings are anagrams of each other.**

import java.util.Arrays;

public class Main {

public static void main(String[] args) {

String str1 = "listen";

String str2 = "silent";

char[] arr1 = str1.toCharArray();

char[] arr2 = str2.toCharArray();

Arrays.sort(arr1);

Arrays.sort(arr2);

boolean isAnagram = Arrays.equals(arr1, arr2);

System.out.println("Are anagrams: " + isAnagram);

}

}

* **Write a program to find the longest word in a given string.**

public class Main {

public static void main(String[] args) {

String str = "Find the longest word in this string";

String[] words = str.split("\\s+");

String longestWord = "";

for (String word : words) {

if (word.length() > longestWord.length()) {

longestWord = word;

}

}

System.out.println("Longest word: " + longestWord);

}

}

* **Write a program to generate all permutations of a given string.**

public class Main {

public static void main(String[] args) {

String str = "ABC";

permute(str, 0, str.length() - 1);

}

private static void permute(String str, int l, int r) {

if (l == r) {

System.out.println(str);

} else {

for (int i = l; i <= r; i++) {

str = swap(str, l, i);

permute(str, l + 1, r);

str = swap(str, l, i); // backtrack

}

}

}

public static String swap(String a, int i, int j) {

char temp;

char[] charArray = a.toCharArray();

temp = charArray[i];

charArray[i] = charArray[j];

charArray[j] = temp;

return String.valueOf(charArray);

}

}

* **Write a program to find the first non-repeated character in a given string.**

import java.util.LinkedHashMap;

import java.util.Map;

public class Main {

public static void main(String[] args) {

String str = "swiss";

char result = firstNonRepeatedCharacter(str);

System.out.println("First non-repeated character: " + result);

}

private static char firstNonRepeatedCharacter(String str) {

Map<Character, Integer> charCountMap = new LinkedHashMap<>();

for (char ch : str.toCharArray()) {

charCountMap.put(ch, charCountMap.getOrDefault(ch, 0) + 1);

}

for (Map.Entry<Character, Integer> entry : charCountMap.entrySet()) {

if (entry.getValue() == 1) {

return entry.getKey();

}

}

return '\0'; // return null character if no unique character is found

}

}

* **Write a program to count the occurrences of a specific character in a string.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

char target = 'o';

int count = 0;

for (char ch : str.toCharArray()) {

if (ch == target) {

count++;

}

}

System.out.println("Occurrences of 'o': " + count);

}

}

* **Write a program to remove duplicate characters from a string.**

public class Main {

public static void main(String[] args) {

String str = "programming";

StringBuilder sb = new StringBuilder();

for (char ch : str.toCharArray()) {

if (sb.indexOf(String.valueOf(ch)) == -1) {

sb.append(ch);

}

}

System.out.println("Without duplicates: " + sb.toString());

}

}

* **Write a program to capitalize the first letter of each word in a string.**

public class Main {

public static void main(String[] args) {

String str = "capitalize each word in this sentence";

String[] words = str.split("\\s+");

StringBuilder sb = new StringBuilder();

for (String word : words) {

sb.append(Character.toUpperCase(word.charAt(0)))

.append(word.substring(1))

.append(" ");

}

System.out.println("Capitalized: " + sb.toString().trim());

}

}

* **Write a program to find the longest substring without repeating characters in a given string.**

public class LongestSubstringWithoutRepeating {

public static int lengthOfLongestSubstring(String s) {

// Array to store the last index of each character

int[] lastIndex = new int[256]; // Assuming ASCII characters

for (int i = 0; i < 256; i++) {

lastIndex[i] = -1; // Initialize all indices to -1

}

int maxLength = 0; // Variable to store the maximum length of substring

int start = 0; // Starting index of the current substring

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

char currentChar = s.charAt(i);

// If the character is already seen and is within the current window

if (lastIndex[currentChar] >= start) {

start = lastIndex[currentChar] + 1; // Move the start to the next character

}

// Update the last seen index of the current character

lastIndex[currentChar] = i;

// Calculate the length of the current substring and update maxLength if needed

maxLength = Math.max(maxLength, i - start + 1);

}

return maxLength; // Return the maximum length of substring found

}

public static void main(String[] args) {

String input = "abcabcbb";

int length = lengthOfLongestSubstring(input);

System.out.println("The length of the longest substring without repeating characters is: " + length);

}

}

* **Write a program to split a string by a specific delimiter.**

public class Main {

public static void main(String[] args) {

String str = "apple,banana,cherry";

String[] fruits = str.split(",");

for (String fruit : fruits) {

System.out.println(fruit);

}

}

}

* **Write a program to reverse a string using StringBuilder.**

public class Main {

public static void main(String[] args) {

StringBuilder sb = new StringBuilder("Hello, World!");

sb.reverse();

System.out.println("Reversed: " + sb.toString());

}

}

* **Write a program to delete a portion of a string using StringBuffer.**

public class Main {

public static void main(String[] args) {

StringBuffer sb = new StringBuffer("Hello, World!");

sb.delete(5, 12);

System.out.println("Deleted: " + sb.toString());

}

}

* **Write a program to count the occurrences of a specific word in a string.**

public class Main {

public static void main(String[] args) {

String str = "this is a test, this is only a test";

String word = "test";

String[] words = str.split("\\s+");

int count = 0;

for (String w : words) {

if (w.equals(word)) {

count++;

}

}

System.out.println("Occurrences of 'test': " + count);

}

}

* **#Write a program to remove all whitespace from a string.**

public class Main {

public static void main(String[] args) {

String str = " Remove all whitespace ";

String noWhitespace = str.replaceAll("\\s", "");

System.out.println("No whitespace: " + noWhitespace);

}

}

* **#Write a program to check if a string contains a specific substring, ignoring case.**

public class Main {

public static void main(String[] args) {

String str = "Hello, World!";

String substr = "world";

boolean containsIgnoreCase = str.toLowerCase().contains(substr.toLowerCase());

System.out.println("Contains 'world' (ignore case): " + containsIgnoreCase);

}

}

* **Write a program to reverse each word in a given string.**

public class Main {

public static void main(String[] args) {

String str = "reverse each word";

String[] words = str.split("\\s+");

StringBuilder result = new StringBuilder();

for (String word : words) {

String reversedWord = new StringBuilder(word).reverse().toString();

result.append(reversedWord).append(" ");

}

System.out.println("Reversed words: " + result.toString().trim());

}

}

* **#Write a program to find all possible substrings of a given string.**

public class Main {

public static void main(String[] args) {

String str = "abc";

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

for (int j = i + 1; j <= str.length(); j++) {

System.out.println(str.substring(i, j));

}

}

}

}