

## 1. Reverse a String

```
public class ReverseString {  
  
    public static void main(String[] args) {  
  
        String str = "Automation";  
  
        StringBuilder reversed = new StringBuilder(str).reverse();  
  
        System.out.println(reversed);  
  
    }  
}
```

## 2. Check for Palindrome

```
public class Palindrome {  
  
    public static void main(String[] args) {  
  
        String str = "madam";  
  
        String reversed = new StringBuilder(str).reverse().toString();  
  
        System.out.println(str.equals(reversed));  
  
    }  
}
```

## 3. Fibonacci Series

```
public class Fibonacci {  
  
    public static void main(String[] args) {  
  
        int n = 10, num1 = 0, num2 = 1;  
  
        System.out.print("Fibonacci Series: " + num1 + ", " + num2);  
  
        for (int i = 2; i < n; i++) {  
  
            int num3 = num1 + num2;  
  
            System.out.print(", " + num3);  
  
            num1 = num2;  
  
            num2 = num3;  
  
        }  
}
```

```
}
```

## 4. Factorial of a Number

```
public class Factorial {  
  
    public static void main(String[] args) {  
  
        int num = 5, factorial = 1;  
  
        for (int i = 1; i <= num; i++) {  
  
            factorial *= i;  
  
        }  
  
        System.out.println(factorial);  
  
    }  
}
```

## 5. Prime Number Check

```
public class PrimeCheck {  
  
    public static void main(String[] args) {  
  
        int num = 11;  
  
        boolean isPrime = true;  
  
        for (int i = 2; i <= Math.sqrt(num); i++) {  
  
            if (num % i == 0) {  
  
                isPrime = false;  
  
                break;  
  
            }  
  
        }  
  
        System.out.println(isPrime);  
  
    }  
}
```

## 6. Count Vowels and Consonants

```
public class VowelConsonantCount {  
  
    public static void main(String[] args) {  
  
        String str = "Automation";
```

```

    int vowels = 0, consonants = 0;

    for (char c : str.toCharArray()) {
        if ("aeiouAEIOU".indexOf(c) != -1) {
            vowels++;
        } else if (Character.isLetter(c)) {
            consonants++;
        }
    }

    System.out.println("Vowels: " + vowels + ", Consonants: " + consonants);
}

```

## 7. Sort an Array

```

import java.util.Arrays;

public class SortArray {

    public static void main(String[] args) {

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

        Arrays.sort(arr);

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

    }
}

```

## 8. Merge Two Arrays

```

import java.util.Arrays;

public class MergeArrays {

    public static void main(String[] args) {

        int[] arr1 = {1, 3, 5};

        int[] arr2 = {2, 4, 6};

        int[] merged = new int[arr1.length + arr2.length];

        System.arraycopy(arr1, 0, merged, 0, arr1.length);

        System.arraycopy(arr2, 0, merged, arr1.length, arr2.length);
    }
}

```

```
        System.out.println(Arrays.toString(merged));
    }
}
```

## 9. Find the Largest Element in an Array

```
public class LargestInArray {

    public static void main(String[] args) {

        int[] arr = {1, 3, 5, 7, 9};

        int largest = arr[0];

        for (int num : arr) {

            if (num > largest) {

                largest = num;

            }

        }

        System.out.println(largest);

    }

}
```

## 10. Remove Duplicates from an Array

```
import java.util.HashSet;

public class RemoveDuplicates {

    public static void main(String[] args) {

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

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

        for (int num : arr) {

            set.add(num);

        }

        System.out.println(set);

    }

}
```

## 11. Check if a Number is Armstrong

```
public class ArmstrongNumber {  
    public static void main(String[] args) {  
        int num = 153, sum = 0, temp = num;  
        while (temp != 0) {  
            int digit = temp % 10;  
            sum += Math.pow(digit, 3);  
            temp /= 10;  
        }  
        System.out.println(num == sum);  
    }  
}
```

## 12. Reverse a Number

```
public class ReverseNumber {  
    public static void main(String[] args) {  
        int num = 12345, reversed = 0;  
        while (num != 0) {  
            reversed = reversed * 10 + num % 10;  
            num /= 10;  
        }  
        System.out.println(reversed);  
    }  
}
```

## 13. Calculate GCD of Two Numbers

```
public class GCD {  
    public static void main(String[] args) {  
        int a = 60, b = 48;  
        while (b != 0) {  
            int temp = b;
```

```

        b = a % b;

        a = temp;
    }

    System.out.println(a);
}
}

```

## 14. Check for Anagram

```

import java.util.Arrays;

public class AnagramCheck {

    public static void main(String[] args) {

        String str1 = "listen", str2 = "silent";

        char[] arr1 = str1.toCharArray();

        char[] arr2 = str2.toCharArray();

        Arrays.sort(arr1);

        Arrays.sort(arr2);

        System.out.println(Arrays.equals(arr1, arr2));

    }

}

```

## 15. Count the Number of Digits in a Number

```

public class CountDigits {

    public static void main(String[] args) {

        int num = 12345;

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

        System.out.println(count);

    }

}

```

## 16. Print the Prime Numbers in a Range

```

public class PrimeInRange {

    public static void main(String[] args) {

```

```

int start = 10, end = 50;

for (int num = start; num <= end; num++) {

    boolean isPrime = true;

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

        if (num % i == 0) {

            isPrime = false;

            break;

        }

    }

    if (isPrime && num > 1) {

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

    }

}

}

```

## 17. Find the Second Largest Element in an Array

```

public class SecondLargest {

    public static void main(String[] args) {

        int[] arr = {12, 35, 1, 10, 34, 1};

        int first = Integer.MIN_VALUE, second = Integer.MIN_VALUE;

        for (int num : arr) {

            if (num > first) {

                second = first;

                first = num;

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

                second = num;

            }

        }

        System.out.println(second);

    }

}

```

```
}
```

## 18. Swap Two Numbers

```
public class SwapNumbers {  
    public static void main(String[] args) {  
        int a = 5, b = 10;  
        a = a + b;  
        b = a - b;  
        a = a - b;  
        System.out.println("a: " + a + ", b: " + b);  
    }  
}
```

## 19. Print the Pascal's Triangle

```
public class PascalsTriangle {  
    public static void main(String[] args) {  
        int rows = 5;  
        for (int i = 0; i < rows; i++) {  
            int num = 1;  
            System.out.format("%" + (rows - i) * 2 + "s", "");  
            for (int j = 0; j <= i; j++) {  
                System.out.format("%4d", num);  
                num = num * (i - j) / (j + 1);  
            }  
            System.out.println();  
        }  
    }  
}
```

## 20. Find the Missing Number in an Array

```
public class MissingNumber {  
    public static void main(String[] args) {
```



```

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

    int n = arr.length + 1;

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

    for (int num : arr) {
        total -= num;
    }

    System.out.println(total);
}
}

```

## 21. Convert Decimal to Binary

```

public class DecimalToBinary {

    public static void main(String[] args) {

        int num = 10;

        String binary = Integer.toBinaryString(num);

        System.out.println(binary);

    }

}

```

## 22. Check for Perfect Number

```

public class PerfectNumber {

    public static void main(String[] args) {

        int num = 28, sum = 0;

        for (int i = 1; i <= num / 2; i++) {

            if (num % i == 0) {

                sum += i;

            }

        }

        System.out.println(num == sum);

    }

}

```

## 23. Implementing a Simple Calculator

```
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 operation (+, -, *, /): ");

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

        double result;

        switch (operation) {

            case '+': result = num1 + num2; break;

            case '-': result = num1 - num2; break;

            case '*': result = num1 * num2; break;

            case '/': result = num1 / num2; break;

            default: throw new IllegalArgumentException("Invalid operation");

        }

        System.out.println("Result: " + result);

    }

}
```

## 24. Find the Sum of Digits of a Number

```
public class SumOfDigits {

    public static void main(String[] args) {

        int num = 12345, sum = 0;

        while (num != 0) {

            sum += num % 10;

            num /= 10;

        }

    }

}
```

```
        System.out.println(sum);
    }
}
```

## 25. Find the Length of a String

```
public class StringLength {
    public static void main(String[] args) {
        String str = "Automation";
        System.out.println(str.length());
    }
}
```

## 26. Check if a String is Empty

```
public class CheckEmptyString {
    public static void main(String[] args) {
        String str = "";
        System.out.println(str.isEmpty());
    }
}
```

## 27. Count the Occurrences of a Character in a String

```
public class CountCharacter {
    public static void main(String[] args) {
        String str = "Automation";
        char ch = 'a';
        int count = 0;
        for (char c : str.toCharArray()) {
            if (c == ch) count++;
        }
        System.out.println(count);
    }
}
```

## 28. Find the First Non-Repeated Character in a String

```
import java.util.LinkedHashMap;

import java.util.Map;

public class FirstNonRepeatedCharacter {

    public static void main(String[] args) {

        String str = "swiss";

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

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

            charCount.put(c, charCount.getOrDefault(c, 0) + 1);

        }

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

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

                System.out.println(entry.getKey());

                break;

            }

        }

    }

}
```

## 29. Remove All Whitespaces from a String

```
public class RemoveWhitespaces {

    public static void main(String[] args) {

        String str = "A u t o m a t i o n ";

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

        System.out.println(result);

    }

}
```

## 30. Find the Common Elements in Two Arrays

```
import java.util.HashSet;

public class CommonElements {
```

```

public static void main(String[] args) {

    int[] arr1 = {1, 2, 3, 4};

    int[] arr2 = {3, 4, 5, 6};

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

    for (int num : arr1) {

        set.add(num);

    }

    for (int num : arr2) {

        if (set.contains(num)) {

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

        }

    }

}

```

## 31. Find the Factorial of a Number using Recursion

```

public class FactorialRecursion {

    public static void main(String[] args) {

        int num = 5;

        System.out.println(factorial(num));

    }

    static int factorial(int n) {

        if (n == 0) return 1;

        return n * factorial(n - 1);

    }

}

```

## 32. Generate Random Numbers

```

import java.util.Random;

public class RandomNumbers {

```

```

public static void main(String[] args) {

    Random random = new Random();

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

        System.out.println(random.nextInt(100)); // Random number between 0-99

    }

}

```

### 33. Check if a Year is Leap Year

```

public class LeapYear {

    public static void main(String[] args) {

        int year = 2024;

        boolean isLeap = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

        System.out.println(isLeap);

    }

}

```

### 34. Find the Sum of First N Natural Numbers

```

public class SumOfNaturalNumbers {

    public static void main(String[] args) {

        int n = 10, sum = n * (n + 1) / 2;

        System.out.println(sum);

    }

}

```

### 35. Implement a Simple Login System

```

import java.util.Scanner;

public class SimpleLogin {

    public static void main(String[] args) {

        String username = "admin";

        String password = "password";

        Scanner scanner = new Scanner(System.in);
    }

}

```

```

System.out.print("Enter username: ");

String inputUsername = scanner.nextLine();

System.out.print("Enter password: ");

String inputPassword = scanner.nextLine();

if (username.equals(inputUsername) && password.equals(inputPassword)) {

    System.out.println("Login successful!");

} else {

    System.out.println("Login failed!");

}

}

}

```

## 36. Check if a String Contains Another String

```

public class StringContains {

    public static void main(String[] args) {

        String str1 = "Automation Testing";

        String str2 = "Testing";

        System.out.println(str1.contains(str2));

    }

}

```

## 37. Find the Maximum Occurring Character in a String

```

import java.util.HashMap;

public class MaxOccurringCharacter {

    public static void main(String[] args) {

        String str = "programming";

        HashMap<Character, Integer> charCount = new HashMap<>();

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

            charCount.put(c, charCount.getOrDefault(c, 0) + 1);

        }

        char maxChar = str.charAt(0);

```

```

    int maxCount = 0;

    for (char c : charCount.keySet()) {
        if (charCount.get(c) > maxCount) {
            maxCount = charCount.get(c);
            maxChar = c;
        }
    }

    System.out.println(maxChar);
}
}

```

## 38. Implementing Bubble Sort

```

public class BubbleSort {

    public static void main(String[] args) {

        int[] arr = {64, 34, 25, 12, 22, 11, 90};

        int n = arr.length;

        for (int i = 0; i < n - 1; i++) {

            for (int j = 0; j < n - i - 1; j++) {

                if (arr[j] > arr[j + 1]) {

                    int temp = arr[j];

                    arr[j] = arr[j + 1];

                    arr[j + 1] = temp;

                }

            }

        }

        for (int num : arr) {

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

        }

    }

}

```



## 39. Implementing Selection Sort

```
public class SelectionSort {  
  
    public static void main(String[] args) {  
  
        int[] arr = {64, 25, 12, 22, 11};  
  
        int n = arr.length;  
  
        for (int i = 0; i < n - 1; i++) {  
  
            int minIndex = i;  
  
            for (int j = i + 1; j < n; j++) {  
  
                if (arr[j] < arr[minIndex]) {  
  
                    minIndex = j;  
  
                }  
  
            }  
  
            int temp = arr[minIndex];  
  
            arr[minIndex] = arr[i];  
  
            arr[i] = temp;  
  
        }  
  
        for (int num : arr) {  
  
            System.out.print(num + " ");  
  
        }  
  
    }  
}
```

Stay tuned with Sonu Madheshiya on LinkedIn <https://www.linkedin.com/in/sonumadheshiya/> for more interesting content.