### 1. Write a for loop that prints the even numbers from 1 to 20.

### public class EvenNumbers {

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

### for (int i = 1; i <= 20; i++) {

### if (i % 2 == 0) {

### System.out.println(i);

### }

### }

### }

### }

### 2. Create a while loop that prompts the user for their flight choice until a valid number is entered.

### import java.util.Scanner;

### public class FlightChoice {

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

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

### int choice = -1;

### while (choice < 0) {

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

### if (scanner.hasNextInt()) {

### choice = scanner.nextInt();

### if (choice >= 0) {

### System.out.println("You entered a valid flight number: " + choice);

### } else {

### System.out.println("Flight number cannot be negative.");

### }

### } else {

### System.out.println("Invalid input. Please enter a number.");

### scanner.next(); // clear the invalid input

### }

### }

### scanner.close();

### }

### }

### 3. Write a Java program that uses a for loop to print the first 10 numbers of the Fibonacci sequence.

### public class Fibonacci {

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

### int n1 = 0, n2 = 1, n3;

### System.out.println(n1);

### System.out.println(n2);

### for (int i = 2; i < 10; i++) {

### n3 = n1 + n2;

### System.out.println(n3);

### n1 = n2;

### n2 = n3;

### }

### }

### }

### 4. Create a Java program using a while loop to calculate the sum of integers from 1 to 100.

### public class SumOfNumbers {

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

### int sum = 0, i = 1;

### while (i <= 100) {

### sum += i;

### i++;

### }

### System.out.println("The sum of numbers from 1 to 100 is: " + sum);

### }

### }

### 5. Implement a do-while loop that prompts the user to enter a number until they enter a negative number.

### import java.util.Scanner;

### public class NegativeNumber {

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

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

### int number;

### do {

### System.out.print("Enter a number (negative to stop): ");

### number = scanner.nextInt();

### } while (number >= 0);

### System.out.println("You entered a negative number. Program stopped.");

### scanner.close();

### }

### }

### 6. Write a Java program that demonstrates the use of the continue statement in a loop.

### public class ContinueExample {

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

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

### if (i % 2 == 0) {

### continue; // Skip even numbers

### }

### System.out.println(i);

### }

### }

### }

### 7. Initialize and print a 2D array of integers in Java.

### public class TwoDArray {

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

### int[][] array = {

### {1, 2, 3},

### {4, 5, 6},

### {7, 8, 9}

### };

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

### for (int j = 0; j < array[i].length; j++) {

### System.out.print(array[i][j] + " ");

### }

### System.out.println();

### }

### }

### }

### 8. Compare and contrast the use of for loops and while loops.

### (Explanation)

### For Loop: Used when the number of iterations is known in advance.

### Example: Iterating through an array.

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

### System.out.println(array[i]);

### }

### While Loop: Used when the number of iterations is not known in advance.

### Example: Prompting the user until a valid input is given.

### while (!valid) {

### System.out.println("Enter valid input");

### }

### 9. Analyze the impact of using a break statement in nested loops.

### (Explanation)

### Break in Nested Loops: A break statement will only exit the loop in which it is written. To exit outer loops, you can use labeled breaks.

### Example:

### public class NestedBreak {

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

### outer:

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

### for (int j = 1; j <= 5; j++) {

### if (j == 3) {

### break outer; // Exit the outer loop

### }

### System.out.println("i = " + i + ", j = " + j);

### }

### }

### }

### }

### 10. Create a method that accepts an array and returns the maximum value using a for loop.

### public class MaxValue {

### public static int findMax(int[] arr) {

### int max = arr[0];

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

### if (arr[i] > max) {

### max = arr[i];

### }

### }

### return max;

### }

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

### int[] numbers = {5, 3, 8, 2, 9, 4};

### System.out.println("Maximum value: " + findMax(numbers));

### }

### }

### 11. Write a Java program that finds the average of numbers stored in an integer array.

### public class ArrayAverage {

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

### int[] numbers = {10, 20, 30, 40, 50};

### int sum = 0;

### for (int num : numbers) {

### sum += num;

### }

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

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

### }

### }

### 12. Write a Java program that sums the elements of a 2D array.

### public class Sum2DArray {

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

### int[][] array = {

### {1, 2, 3},

### {4, 5, 6},

### {7, 8, 9}

### };

### int sum = 0;

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

### for (int j = 0; j < array[i].length; j++) {

### sum += array[i][j];

### }

### }

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

### }

### }

### 13. Demonstrate how to find the minimum and maximum values in a given array.

### public class MinMaxValue {

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

### int[] numbers = {15, 42, 7, 29, 85, 10};

### int min = numbers[0];

### int max = numbers[0];

### for (int num : numbers) {

### if (num < min) {

### min = num;

### }

### if (num > max) {

### max = num;

### }

### }

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

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

### }

### }

### 14. Design a Java program that merges two sorted arrays into a single sorted array.

### import java.util.Arrays;

### public class MergeSortedArrays {

### public static int[] mergeArrays(int[] arr1, int[] arr2) {

### int[] mergedArray = new int[arr1.length + arr2.length];

### int i = 0, j = 0, k = 0;

### while (i < arr1.length && j < arr2.length) {

### if (arr1[i] <= arr2[j]) {

### mergedArray[k++] = arr1[i++];

### } else {

### mergedArray[k++] = arr2[j++];

### }

### }

### while (i < arr1.length) {

### mergedArray[k++] = arr1[i++];

### }

### while (j < arr2.length) {

### mergedArray[k++] = arr2[j++];

### }

### return mergedArray;

### }

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

### int[] arr1 = {1, 3, 5, 7};

### int[] arr2 = {2, 4, 6, 8};

### int[] result = mergeArrays(arr1, arr2);

### System.out.println("Merged Sorted Array: " + Arrays.toString(result));

### }

### }

### 15. Write a Java program to reverse the array.

### import java.util.Arrays;

### public class ReverseArray {

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

### int[] arr = {10, 20, 30, 40, 50};

### int n = arr.length;

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

### int temp = arr[i];

### arr[i] = arr[n - 1 - i];

### arr[n - 1 - i] = temp;

### }

### System.out.println("Reversed Array: " + Arrays.toString(arr));

### }

### }

### 16. Find the second largest element in an array.

### public class SecondLargest {

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

### int[] arr = {5, 10, 20, 8, 15};

### int largest = Integer.MIN\_VALUE;

### int secondLargest = Integer.MIN\_VALUE;

### for (int num : arr) {

### if (num > largest) {

### secondLargest = largest;

### largest = num;

### } else if (num > secondLargest && num != largest) {

### secondLargest = num;

### }

### }

### System.out.println("Second Largest Element: " + secondLargest);

### }

### }

### 17. Find the first even number in a list and break the loop when it is found.

### public class FirstEvenNumber {

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

### int[] arr = {3, 5, 7, 10, 15};

### for (int num : arr) {

### if (num % 2 == 0) {

### System.out.println("First even number: " + num);

### break;

### }

### }

### }

### }

### 18. Print all odd numbers from 1 to 20, using continue to skip even numbers.

### public class OddNumbers {

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

### for (int i = 1; i <= 20; i++) {

### if (i % 2 == 0) {

### continue; // Skip even numbers

### }

### System.out.println(i);

### }

### }

### }

### 19. Prompt the user to enter numbers until they enter a negative number.

### import java.util.Scanner;

### public class PromptUntilNegative {

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

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

### Int number;

### do {

### System.out.print("Enter a number (negative to stop): ");

### number = scanner.nextInt();

### } while (number >= 0);

### System.out.println("Program ended.");

### scanner.close();

### }

### }

### 20. Print a multiplication table but skip the multiplication by 5.

### public class MultiplicationTable {

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

### int number = 7; // Example number for multiplication table

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

### if (i == 5) {

### continue; // Skip multiplication by 5

### }

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

### }

### }

### }

### 21. Program counts from 1 to 10 but breaks when it reaches 6.

### public class BreakAtSix {

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

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

### if (i == 6) {

### break; // Exit loop when i is 6

### }

### System.out.println(i);

### }

### }

### }

### 22. Program prints numbers from 1 to 10 but skips the number 5.

### public class SkipFive {

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

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

### if (i == 5) {

### continue; // Skip 5

### }

### System.out.println(i);

### }

### }

### }

### 23. Develop a program that checks whether a given number is prime or not.

### import java.util.Scanner;

### public class PrimeCheck {

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

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

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

### int number = scanner.nextInt();

### boolean isPrime = true;

### if (number <= 1) {

### isPrime = false; // 1 and numbers less than 1 are not prime

### } else {

### for (int i = 2; i <= Math.sqrt(number); 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.");

### }

### scanner.close();

### }

### }

### 24. Create a program that reverses the digits of a given integer.

### import java.util.Scanner;

### public class ReverseDigits {

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

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

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

### int number = scanner.nextInt();

### int reversed = 0;

### while (number != 0) {

### int digit = number % 10;

### reversed = reversed \* 10 + digit;

### number /= 10;

### }

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

### scanner.close();

### }

### }

### 25. Write a program that prints the multiplication table for a given number.

### import java.util.Scanner;

### public class MultiplicationTable {

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

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

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

### int number = scanner.nextInt();

### System.out.print("Enter the range: ");

### int range = scanner.nextInt();

### for (int i = 1; i <= range; i++) {

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

### }

### scanner.close();

### }

### }

### 26. Write a program that counts the number of vowels and consonants in a given string.

### import java.util.Scanner;

### public class CountVowelsConsonants {

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

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

### System.out.print("Enter a string: ");

### String input = scanner.nextLine().toLowerCase();

### int vowels = 0, consonants = 0;

### for (char c : input.toCharArray()) {

### if ("aeiou".indexOf(c) != -1) {

### vowels++;

### } else if (Character.isLetter(c)) {

### consonants++;

### }

### }

### System.out.println("Vowels: " + vowels);

### System.out.println("Consonants: " + consonants);

### scanner.close();

### }

### }

### 27. Print the pattern as given:

### 1 1 1 1 1

### 1 1 1 1

### 1 1 1

### 1 1

### 1

### public class PrintPattern {

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

### for (int i = 5; i >= 1; i--) {

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

### System.out.print("1 ");

### }

### System.out.println();

### }

### }

### }

### 28. Develop a program for collecting customer ratings and calculating the average.

### import java.util.Scanner;

### public class CustomerRatings {

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

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

### int totalRatings = 0, numberOfRatings = 0, rating;

### while (true) {

### System.out.print("Enter a rating (1-5) or 0 to stop: ");

### rating = scanner.nextInt();

### if (rating == 0) {

### break;

### } else if (rating >= 1 && rating <= 5) {

### totalRatings += rating;

### numberOfRatings++;

### } else {

### System.out.println("Invalid rating. Please enter a number between 1 and 5.");

### }

### }

### if (numberOfRatings > 0) {

### double average = (double) totalRatings / numberOfRatings;

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

### System.out.println("Number of ratings: " + numberOfRatings);

### } else {

### System.out.println("No ratings received.");

### }

### scanner.close();

### }

### }

### 29. Develop a program to track monthly expenses.

### import java.util.Scanner;

### public class MonthlyExpenses {

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

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

### double totalExpenses = 0;

### String category;

### while (true) {

### System.out.print("Enter expense category (type 'done' to finish): ");

### category = scanner.next();

### if (category.equalsIgnoreCase("done")) {

### break;

### }

### System.out.print("Enter expense amount for " + category + ": ");

### double amount = scanner.nextDouble();

### totalExpenses += amount;

### }

### System.out.println("Total expenses for the month: $" + totalExpenses);

### scanner.close();

### }

### }

### 30. Develop a password validation system.

### import java.util.Scanner;

### public class PasswordValidator {

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

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

### String password;

### while (true) {

### System.out.print("Create a password (at least 8 characters, includes a number and a special character): ");

### password = scanner.next();

### if (password.length() >= 8 &&

### password.matches(".\*\\d.\*") &&

### password.matches(".\*[!@#$%^&\*()].\*")) {

### System.out.println("Password created successfully.");

### break;

### } else {

### System.out.println("Invalid password. Try again.");

### }

### }

### scanner.close();

### }

### }

### 31. Create a fitness app to log daily steps.

### import java.util.Scanner;

### public class FitnessApp {

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

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

### int[] steps = new int[7];

### int totalSteps = 0;

### for (int i = 0; i < 7; i++) {

### System.out.print("Enter steps for day " + (i + 1) + ": ");

### steps[i] = scanner.nextInt();

### totalSteps += steps[i];

### }

### double averageSteps = (double) totalSteps / 7;

### System.out.println("Total steps taken: " + totalSteps);

### System.out.println("Average steps per day: " + averageSteps);

### scanner.close();

### }

### }

### 32. Develop a temperature conversion tool.

### import java.util.Scanner;

### public class TemperatureConverter {

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

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

### int choice;

### do {

### System.out.println("1. Convert Celsius to Fahrenheit");

### System.out.println("2. Convert Fahrenheit to Celsius");

### System.out.println("3. Exit");

### System.out.print("Choose an option: ");

### choice = scanner.nextInt();

### switch (choice) {

### case 1:

### System.out.print("Enter temperature in Celsius: ");

### double celsius = scanner.nextDouble();

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

### System.out.println("Temperature in Fahrenheit: " + fahrenheit);

### break;

### case 2:

### System.out.print("Enter temperature in Fahrenheit: ");

### fahrenheit = scanner.nextDouble();

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

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

### break;

### case 3:

### System.out.println("Exiting...");

### break;

### default:

### System.out.println("Invalid choice. Try again.");

### }

### } while (choice != 3);

### scanner.close();

### }

### }

### 33. Implement a simple banking system.

### import java.util.Scanner;

### public class BankingSystem {

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

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

### double balance = 0;

### String transactionHistory = "";

### while (true) {

### System.out.println("\n1. Deposit");

### System.out.println("2. Withdraw");

### System.out.println("3. Check Balance");

### System.out.println("4. Exit");

### System.out.print("Choose an option: ");

### int choice = scanner.nextInt();

### switch (choice) {

### case 1:

### System.out.print("Enter amount to deposit: ");

### double deposit = scanner.nextDouble();

### balance += deposit;

### transactionHistory += "Deposited: $" + deposit + "\n";

### break;

### case 2:

### System.out.print("Enter amount to withdraw: ");

### double withdraw = scanner.nextDouble();

### if (withdraw > balance) {

### System.out.println("Insufficient balance!");

### } else {

### balance -= withdraw;

### transactionHistory += "Withdrew: $" + withdraw + "\n";

### }

### break;

### case 3:

### System.out.println("Current balance: $" + balance);

### break;

### case 4:

### System.out.println("Final balance: $" + balance);

### System.out.println("Transaction History:\n" + transactionHistory);

### scanner.close();

### return;

### default:

### System.out.println("Invalid option. Try again.");

### }

### }

### }

### }

### 34. Create a program to input grades for students and calculate statistics.

### import java.util.Scanner;

### public class GradeStatistics {

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

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

### int grade, sum = 0, count = 0, highest = Integer.MIN\_VALUE, passCount = 0;

### while (true) {

### System.out.print("Enter a grade (-1 to stop): ");

### grade = scanner.nextInt();

### if (grade == -1) break;

### sum += grade;

### count++;

### if (grade > highest) highest = grade;

### if (grade >= 40) passCount++; // Assuming pass threshold is 40

### }

### if (count > 0) {

### double average = (double) sum / count;

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

### System.out.println("Highest grade: " + highest);

### System.out.println("Number of students passed: " + passCount);

### } else {

### System.out.println("No grades entered.");

### }

### scanner.close();

### }

### }

### 35. Design a shopping cart application.

### import java.util.Scanner;

### public class ShoppingCart {

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

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

### double totalAmount = 0;

### StringBuilder itemsPurchased = new StringBuilder();

### while (true) {

### System.out.print("Enter item name (type 'checkout' to finish): ");

### String item = scanner.next();

### if (item.equalsIgnoreCase("checkout")) break;

### System.out.print("Enter item price: ");

### double price = scanner.nextDouble();

### totalAmount += price;

### itemsPurchased.append(item).append(" ($").append(price).append(")\n");

### }

### System.out.println("\nItems Purchased:");

### System.out.println(itemsPurchased);

### System.out.println("Total Amount Due: $" + totalAmount);

### scanner.close();

### }

### }

### 36. Write a program to calculate total sales and commission for salespeople.

### import java.util.Scanner;

### public class SalesCommission {

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

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

### double totalSales = 0;

### int count = 0;

### while (true) {

### System.out.print("Enter sales figure (negative number to stop): ");

### double sales = scanner.nextDouble();

### if (sales < 0) break;

### totalSales += sales;

### count++;

### }

### if (count > 0) {

### double averageSales = totalSales / count;

### System.out.println("Total Sales: $" + totalSales);

### System.out.println("Average Sales per Salesperson: $" + averageSales);

### } else {

### System.out.println("No sales data entered.");

### }

### scanner.close();

### }

### }

### 37. Write a Java program to reverse a string.

### import java.util.Scanner;

### public class ReverseString {

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

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

### System.out.print("Enter a string: ");

### String input = scanner.nextLine();

### String reversed = new StringBuilder(input).reverse().toString();

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

### scanner.close();

### }

### }

### 38. Check if a string is a palindrome.

### import java.util.Scanner;

### public class PalindromeCheck {

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

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

### System.out.print("Enter a string: ");

### String input = scanner.nextLine();

### String reversed = new StringBuilder(input).reverse().toString();

### if (input.equalsIgnoreCase(reversed)) {

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

### } else {

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

### }

### scanner.close();

### }

### }

### 39. Identify and count the occurrences of each character in a string.

### import java.util.HashMap;

### import java.util.Scanner;

### public class CharacterCount {

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

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

### System.out.print("Enter a string: ");

### String input = scanner.nextLine();

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

### for (char c : input.toCharArray()) {

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

### }

### System.out.println("Character occurrences:");

### for (char key : charCount.keySet()) {

### System.out.println(key + ": " + charCount.get(key));

### }

### scanner.close();

### }

### }

### 40. Remove duplicate characters from a string while maintaining order.

### import java.util.LinkedHashSet;

### import java.util.Scanner;

### public class RemoveDuplicates {

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

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

### System.out.print("Enter a string: ");

### String input = scanner.nextLine();

### LinkedHashSet<Character> uniqueChars = new LinkedHashSet<>();

### for (char c : input.toCharArray()) {

### uniqueChars.add(c);

### }

### StringBuilder result = new StringBuilder();

### for (char c : uniqueChars) {

### result.append(c);

### }

### System.out.println("String after removing duplicates: " + result);

### scanner.close();

### }

### }

### ****41. Total Sales and Commission Calculation****

import java.util.Scanner;

public class SalesCommission {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

double totalSales = 0;

int salespersonCount = 0;

while (true) {

System.out.print("Enter sales amount for salesperson (negative number to stop): ");

double sales = scanner.nextDouble();

if (sales < 0) break;

totalSales += sales;

salespersonCount++;

}

if (salespersonCount > 0) {

double averageSales = totalSales / salespersonCount;

System.out.println("Total Sales: $" + totalSales);

System.out.println("Average Sales per Salesperson: $" + averageSales);

} else {

System.out.println("No sales data entered.");

}

scanner.close();

}

}

### ****42. Reverse a String****

import java.util.Scanner;

public class ReverseString {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a string to reverse: ");

String input = scanner.nextLine();

String reversed = new StringBuilder(input).reverse().toString();

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

scanner.close();

}

}

### ****43. Check if a String is a Palindrome****

import java.util.Scanner;

public class PalindromeCheck {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String input = scanner.nextLine();

String reversed = new StringBuilder(input).reverse().toString();

if (input.equals(reversed)) {

System.out.println("The string is a palindrome.");

} else {

System.out.println("The string is not a palindrome.");

}

scanner.close();

}

}

### ****44. Count Character Occurrences in a String****

import java.util.HashMap;

import java.util.Scanner;

public class CharacterCount {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String input = scanner.nextLine();

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

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

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

}

System.out.println("Character counts: " + charCountMap);

scanner.close();

}

}

### ****45. Reverse a String Without Using Built-in Reverse Method****

import java.util.Scanner;

public class ReverseStringManual {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a string to reverse: ");

String input = scanner.nextLine();

String reversed = "";

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

reversed += input.charAt(i);

}

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

scanner.close();

}

}

### ****46. Check if a String is a Palindrome (Method-Based)****

public class PalindromeMethod {

public static boolean isPalindrome(String str) {

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

return str.equals(reversed);

}

public static void main(String[] args) {

String input = "racecar";

System.out.println("Is '" + input + "' a palindrome? " + isPalindrome(input));

}

}

### ****47. Count Vowels and Consonants in a String****

public class CountVowelsConsonants {

public static void main(String[] args) {

String input = "Hello World";

int vowels = 0, consonants = 0;

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

if (c >= 'a' && c <= 'z') {

if ("aeiou".indexOf(c) != -1) {

vowels++;

} else {

consonants++;

}

}

}

System.out.println("Vowels: " + vowels);

System.out.println("Consonants: " + consonants);

}

}

### ****48. Capitalize the First Letter of Each Word in a String****

public class CapitalizeWords {

public static String capitalize(String str) {

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

StringBuilder capitalized = new StringBuilder();

for (String word : words) {

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

.append(word.substring(1)).append(" ");

}

return capitalized.toString().trim();

}

public static void main(String[] args) {

String input = "hello world";

System.out.println("Capitalized String: " + capitalize(input));

}

}

### ****49. Check if Two Strings are Anagrams****

import java.util.Arrays;

public class AnagramCheck {

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

char[] charArray1 = str1.toCharArray();

char[] charArray2 = str2.toCharArray();

Arrays.sort(charArray1);

Arrays.sort(charArray2);

return Arrays.equals(charArray1, charArray2);

}

public static void main(String[] args) {

String str1 = "listen";

String str2 = "silent";

System.out.println("Are '" + str1 + "' and '" + str2 + "' anagrams? " + areAnagrams(str1, str2));

}

}

### ****50. Remove Duplicate Characters from a String****

public class RemoveDuplicates {

public static String removeDuplicates(String str) {

StringBuilder result = new StringBuilder();

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

if (result.indexOf(String.valueOf(c)) == -1) {

result.append(c);

}

}

return result.toString();

}

public static void main(String[] args) {

String input = "programming";

System.out.println("String after removing duplicates: " + removeDuplicates(input));

}

}

### ****51. Find the First Non-Repeating Character in a String****

import java.util.LinkedHashMap;

import java.util.Map;

public class FirstNonRepeating {

public static char firstNonRepeatingChar(String str) {

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

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

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

}

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

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

return entry.getKey();

}

}

return '\0'; // Return null character if no non-repeating character is found

}

public static void main(String[] args) {

String input = "swiss";

char result = firstNonRepeatingChar(input);

System.out.println("First non-repeating character: " + (result != '\0' ? result : "None"));

}

}

### ****52. Compress a String Using Counts of Repeated Characters****

public class StringCompression {

public static String compressString(String str) {

StringBuilder compressed = new StringBuilder();

int count = 1;

for (int i = 0; i < str.length() - 1; i++) {

if (str.charAt(i) == str.charAt(i + 1)) {

count++;

} else {

compressed.append(str.charAt(i)).append(count);

count = 1;

}

}

compressed.append(str.charAt(str.length() - 1)).append(count);

return compressed.length() < str.length() ? compressed.toString() : str;

}

public static void main(String[] args) {

String input = "aabcccccaaa";

System.out.println("Compressed String: " + compressString(input));

}

}

### ****53. Append "World" to a StringBuffer****

public class AppendStringBuffer {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer("Hello");

buffer.append(" World");

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

}

}

### ****54. Insert "Beautiful" into a StringBuffer****

public class InsertStringBuffer {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer("Hello World");

buffer.insert(6, "Beautiful ");

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

}

}

### ****55. Reverse a StringBuffer****

public class ReverseStringBuffer {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer("Java Programming");

buffer.reverse();

System.out.println("Reversed StringBuffer: " + buffer);

}

}

### ****56. Delete a Substring from a StringBuffer****

public class DeleteSubstring {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer("Hello World");

buffer.delete(6, 11); // Removes "World"

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

}

}

### ****57. Reverse Contents of a StringBuffer****

public class ReverseBufferContent {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer("Java Programming");

buffer.reverse();

System.out.println("Reversed StringBuffer: " + buffer);

}

}

### ****58. Remove "World" from a StringBuffer****

public class RemoveWord {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer("Hello World");

int start = buffer.indexOf("World");

int end = start + "World".length();

buffer.delete(start, end);

System.out.println("Modified StringBuffer: " + buffer);

}

}

### ****59. Replace "Java" with "Python" in a StringBuffer****

public class ReplaceWord {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer("I love Java programming");

int start = buffer.indexOf("Java");

int end = start + "Java".length();

buffer.replace(start, end, "Python");

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

}

}

### ****60. Check Initial and New Capacity of a StringBuffer****

public class BufferCapacity {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer();

System.out.println("Initial Capacity: " + buffer.capacity());

buffer.append("This is a test string that exceeds the initial capacity.");

System.out.println("New Capacity: " + buffer.capacity());

}

}

### ****61. Convert a StringBuffer to a String****

public class BufferToString {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer("Hello World");

String result = buffer.toString();

System.out.println("Converted String: " + result);

}

}

### ****62. Count Vowels in a StringBuffer****

public class CountVowelsBuffer {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer("Hello World");

int vowelCount = 0;

for (char c : buffer.toString().toLowerCase().toCharArray()) {

if ("aeiou".indexOf(c) != -1) {

vowelCount++;

}

}

System.out.println("Number of Vowels: " + vowelCount);

}

}

### ****63. Trim Whitespace in a StringBuffer****

public class TrimWhitespace {

public static void main(String[] args) {

StringBuffer buffer = new StringBuffer(" Hello World ");

String trimmed = buffer.toString().trim();

System.out.println("Trimmed String: " + trimmed);

}

}

### ****64. Merge Two StringBuffers****

public class MergeBuffers {

public static void main(String[] args) {

StringBuffer buffer1 = new StringBuffer("Hello");

StringBuffer buffer2 = new StringBuffer("World");

StringBuffer merged = new StringBuffer(buffer1).append(" ").append(buffer2);

System.out.println("Merged StringBuffer: " + merged);

}

}