1. Write a program to reverse a word using loop? (Not to use inbuilt functions)

Sample Input:

String: TEMPLE

Sample Output:

Reverse String: ELPMET

ANSWER:

public class ReverseWord

{

public static void main(String[] args)

{

String inputString = "TEMPLE";

String outputString = reverseWord(inputString);

System.out.println("Original String: " + inputString);

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

}

private static String reverseWord(String inputStr)

{

char[] charArray = inputStr.toCharArray();

int start = 0;

int end = charArray.length - 1;

while (start < end)

{

char temp = charArray[start];

charArray[start] = charArray[end];

charArray[end] = temp;

start++;

end--;

}

return new String(charArray);

}

}

2.Write a program to convent the given string to integer?

Sample Input:

String: 1234

Sample Output:

Out put String: 1234

ANSWER :

public class StringToIntegerConverter

{

public static void main(String[] args)

{

String inputString = "1234";

try {

int outputInteger = Integer.parseInt(inputString);

System.out.println("Output Integer: " + outputInteger);

} catch (NumberFormatException e) {

System.out.println("Invalid input. Please provide a valid numeric string.");

}

}

}

3.Write a program to check the entered user name is valid or not. Get both the inputs from the user.

ANSWER:

public class StringToIntegerConverter

{

public static void main(String[] args)

{

String inputString = "1234";

try {

int outputInteger = Integer.parseInt(inputString);

System.out.println("Output Integer: " + outputInteger);

}

catch (NumberFormatException e) {

System.out.println("Invalid input. Please provide a valid numeric string.");

}

}

}

4.Write a program that would sort a list of names in alphabetical order Ascending or Descending, choice get from the user?

Sample Input:

Banana

Carrot

Radish

Apple

Jack

Order(A/D) : A

Sample Output:

Apple

Banana

Carrot

Jack

Radish

ANSWER:

import java.util.Scanner;

public class SortNames {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

ArrayList<String>namesList = new ArrayList<>();

System.out.println("Enter names (Enter an empty string to finish):");

String name;

do {

System.out.print("Name: ");

name = scanner.nextLine();

if (!name.isEmpty()) {

namesList.add(name);

}

} while (!name.isEmpty());

System.out.print("Enter sorting order (A for Ascending, D for Descending): ");

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

sortNames(namesList, sortOrder);

System.out.println("Sorted Names:");

for (String sortedName :namesList) {

System.out.println(sortedName);

}

scanner.close();

}

private static void sortNames(ArrayList<String> names, char sortOrder) {

if (sortOrder == 'A' || sortOrder == 'a') {

Collections.sort(names);

} else if (sortOrder == 'D' || sortOrder == 'd') {

Collections.sort(names, Collections.reverseOrder());

}

}

}

5.Write a program to print the special characters separately and print number of Special characters in the line?

ANSWER:

import java.util.Scanner;

public class SpecialCharacterCounter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a line of text: ");

String inputLine = scanner.nextLine();

System.out.println("Special characters in the line:");

int specialCharacterCount = 0;

for (char character :inputLine.toCharArray()) {

if (isSpecialCharacter(character)) {

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

specialCharacterCount++;

}

}

System.out.println("\nNumber of special characters: " + specialCharacterCount);

scanner.close();

}

private static booleanisSpecialCharacter(char ch) {

return !Character.isLetterOrDigit(ch) && !Character.isWhitespace(ch);

}

}

6,Write a program to print the number of vowels in the given statement?

Sample Input:

Saveetha School of Engineering

Sample Output:

Number o vowels = 12

ANSWER:

import java.util.Scanner;

public class VowelCounter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String inputStatement = scanner.nextLine();

int vowelCount = countVowels(inputStatement);

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

scanner.close();

}

private static int countVowels(String inputStr) {

int count = 0;

for (char character :inputStr.toLowerCase().toCharArray()) {

if (isVowel(character)) {

count++;

}

}

return count;

}

private static booleanisVowel(char character) {

return "aeiou".indexOf(character) != -1;

}

}

7.Write a program to print consonants and vowels separately in the given word

Sample Input:

Given Word: Engineering

Sample Output:

Consonants: n g n r n g

Vowels: e i e ei

ANSWER:

import java.util.Scanner;

public class SeparateConsonantsAndVowels {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String inputWord = scanner.nextLine();

String consonants = getConsonants(inputWord);

String vowels = getVowels(inputWord);

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

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

scanner.close();}

private static String getConsonants(String inputStr) {

StringBuilder consonants = new StringBuilder();

for (char character :inputStr.toCharArray()) {

if (isConsonant(character)) {

consonants.append(character).append(" ");

}

}

return consonants.toString().trim();

}

private static String getVowels(String inputStr) {

StringBuilder vowels = new StringBuilder();

for (char character :inputStr.toCharArray()) {

if (isVowel(character)) {

vowels.append(character).append(" ");

}

}

return vowels.toString().trim();}

private static booleanisConsonant(char character) {

return Character.isLetter(character) && !isVowel(character);

}

private static booleanisVowel(char character) {

return "aeiouAEIOU".indexOf(character) != -1;

}

}

8.Write a program that finds whether a given character is present in a string or not. In case it is present it prints the index at which it is present. Do not use built-in find functions to search the character.

Sample Input:

Enter the string: I am a programmer

Enter the character to be searched: p

Sample Output:

P is found in string at index: 8

Note: Check for non available Character in the given statement as Hidden Test case.

ANSWER:

import java.util.Scanner;

public class CharacterSearch {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String inputString = scanner.nextLine();

System.out.print("Enter the character to be searched: ");

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

findAndPrintIndex(inputString, searchChar);

scanner.close()

;}

private static void findAndPrintIndex(String inputStr, char searchChar) {

boolean found = false;

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

if (inputStr.charAt(i) == searchChar) {

System.out.println(searchChar + " is found in string at index: " + i);

found = true;

break;

}

}

if (!found) {

System.out.println(searchChar + " is not present in the given string.");

}

}

}

9.Write a program to arrange the letters of the word alphabetically in reverse order

Sample Input:

Enter the word: MOSQUE

Sample Output:

Alphabetical Order: U S Q O M E

Test Case:

1. HYPOTHECATION
2. MATRICULATION
3. MANIPULATION

ANSWER:

import java.util.Scanner;

public class CharacterSearch {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String inputString = scanner.nextLine();

System.out.print("Enter the character to be searched: ");

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

findAndPrintIndex(inputString, searchChar);

scanner.close();

}

private static void findAndPrintIndex(String inputStr, char searchChar) {

boolean found = false;

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

if (inputStr.charAt(i) == searchChar) {

System.out.println(searchChar + " is found in string at index: " + i);

found = true;

break;

}

}

if (!found) {

System.out.println(searchChar + " is not present in the given string.");

}

}

}

10.Write a program that accepts a string from user and displays the same string after removing vowels from it.

Sample Input & Output:

Enter a string: we can play the game

The string without vowels is: w cn ply thgm

ANSWER:

import java.util.Scanner;

public class RemoveVowels {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String inputString = scanner.nextLine();

String withoutVowels = removeVowels(inputString);

System.out.println("The string without vowels is: " + withoutVowels);

scanner.close();

}

private static String removeVowels(String inputStr) {

StringBuilder result = new StringBuilder();

for (char character :inputStr.toCharArray()) {

if (!isVowel(character)) {

result.append(character);

}

}

return result.toString();

}

private static booleanisVowel(char character) {

return "aeiouAEIOU".indexOf(character) != -1;

}

}

11.Write a program for matrix multiplication?

Sample Input:

Mat1 = 1 2

5 3

Mat2 = 2 3

4 1

Sample Output:

Mat Sum = 10 5

22 18

ANSWER:

import java.util.Scanner;

public class MatrixMultiplication {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the dimensions of the first matrix:");

System.out.print("Rows: ");

int rows1 = scanner.nextInt();

System.out.print("Columns: ");

int cols1 = scanner.nextInt();

System.out.println("Enter the dimensions of the second matrix:");

System.out.print("Rows: ");

int rows2 = scanner.nextInt();

System.out.print("Columns: ");

int cols2 = scanner.nextInt();

if (cols1 != rows2) {

System.out.println("Matrix multiplication is not possible with the given dimensions.");

} else {

int[][] matrix1 = inputMatrix("Enter elements for the first matrix:", rows1, cols1, scanner);

int[][] matrix2 = inputMatrix("Enter elements for the second matrix:", rows2, cols2, scanner);

int[][] resultMatrix = multiplyMatrices(matrix1, matrix2);

System.out.println("Matrix Multiplication Result:");

displayMatrix(resultMatrix);

}

scanner.close();

}

private static int[][] inputMatrix(String message, int rows, int cols, Scanner scanner) {

System.out.println(message);

int[][] matrix = new int[rows][cols];

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

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

System.ut.print("Enter element at position (" + (i + 1) + ", " + (j + 1) + "): ");

matrix[i] [j] = scanner.nextInt();

}

}

return matrix;

}

private static int[][] multiplyMatrices(int[][] matrix1, int[][] matrix2) {

int rows1 = matrix1.length;

int cols1 = matrix1[0].length;

int cols2 = matrix2[0].length;

int[][] resultMatrix = new int[rows1][cols2];

forint i = 0; i< rows1; i++) {

for (int j = 0; j < cols2; j++) {

for (int k = 0; k < cols1; k++) {

resultMatrix[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

}

return resultMatrix;

}

private static void displayMatrix(int[][] matrix) {

for (int[] row : matrix) {

for (int element : row) {

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

}

System.out.println();

}

}

}

12.Write a program for matrix addition?

Sample Input:

Mat1 = 1 2

5 3

Mat2 = 2 3

4 1

Sample Output:

Mat Sum = 3 5

9 4

import java.util.Scanner;

ANSWER:

public class MatrixAddition {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the dimensions of the matrices:");

System.out.print("Rows: ");

int rows = scanner.nextInt();

System.out.print("Columns: ");

int[][] matrix1 = inputMatrix("Enter elements for the first matrix:", rows, cols, scanner);

int[][] matrix2 = inputMatrix("Enter elements for the second matrix:", rows, cols, scanner);

int[][] sumMatrix = addMatrices(matrix1, matrix2);

System.out.println("Matrix Addition Result:");

displayMatrix(sumMatrix);

scanner.close();

}

private static int[][] inputMatrix(String message, int rows, int cols, Scanner scanner) {

System.out.println(message);

int[][] matrix = new int[rows][cols];

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

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

System.out.print("Enter element at position (" + (i + 1) + ", " + (j + 1) + "): ");

matrix[i][j] = scanner.nextInt();

}

}

return matrix;

}

private static int[][] addMatrices(int[][] matrix1, int[][] matrix2) {

int rows = matrix1.length;

int cols = matrix1[0].length;

int[][] sumMatrix = new int[rows][cols];

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

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

sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

return sumMatrix;

}

private static void displayMatrix(int[][] matrix) {

for (int[] row : matrix) {

for (int element : row) {

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

}

System.out.println();

}

}

}

13.Write a program for Merge two sorted arrays using Array list

Input: arr1[] = { 1, 3, 4, 5}, arr2[] = {2, 4, 6, 8}

Output: arr3[] = {1, 2, 3, 4, 4, 5, 6, 8}

ANSWER:

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

public class MergeSortedArrays {

public static void main(String[] args) {

int[] arr1 = {1, 3, 4, 5};

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

List<Integer>mergedList = mergeArrays(arr1, arr2);

Integer[] mergedArray = mergedList.toArray(new Integer[0]);

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

}

private static List<Integer>mergeArrays(int[] arr1, int[] arr2) {

List<Integer>mergedList = new ArrayList<>();

int i = 0, j = 0;

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

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

mergedList.add(arr1[i]);

i++;

} else if (arr1[i] > arr2[j]) {

mergedList.add(arr2[j]);

j++;

} else {

mergedList.add(arr1[i]);

mergedList.add(arr2[j]);

i++;

j++;

}

}

while (i< arr1.length) {

mergedList.add(arr1[i]);

i++;

}

while (j < arr2.length) {

mergedList.add(arr2[j]);

j++;

}

return mergedList;

}

}

14.Find the Mean, Median, Mode of the array of numbers?

Sample Input;:

Array of elements = {16, 18, 27, 16, 23, 21, 19}

Sample Output:  
Mean = 20

Median = 19

Mode = 16

Test cases:

1. Array of elements = {26, 28, 37, 26, 33, 31, 29}

2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}

3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}

4. Array of elements = {200, 180, 180, 270, 160, 270, 270, 190, 200}

5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100, 100}

ANSWER:

import java.util.Arrays;

import java.util.HashMap;

import java.util.Map;

public class MeanMedianMode {

public static void main(String[] args) {

double[] array = {16, 18, 27, 16, 23, 21, 19};

double mean = calculateMean(array);

double median = calculateMedian(array);

double mode = calculateMode(array);

System.out.println("Mean = " + mean);

System.out.println("Median = " + median);

System.out.println("Mode = " + mode);

double[] testArray1 = {26, 28, 37, 26, 33, 31, 29};

double[] testArray2 = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, 0.19};

double[] testArray3 = {0, 160, 180, 270, 160, 230, 210, 190, 0};

double[] testArray4 = {200, 180, 180, 270, 160, 270, 270, 190, 200};

double[] testArray5 = {100, 100, 100, 100, 100, 100, 100, 100, 100};

System.out.println("\nTest Case 1:");

displayStatistics(testArray1);

System.out.println("\nTest Case 2:");

displayStatistics(testArray2);

System.out.println("\nTest Case 3:");

displayStatistics(testArray3);

System.out.println("\nTest Case 4:");

displayStatistics(testArray4);

System.out.println("\nTest Case 5:");

displayStatistics(testArray5);

}

private static double calculateMean(double[] array) {

double sum = 0;

for (double num : array) {

sum += num;

}

return sum / array.length;

}

private static double calculateMedian(double[] array)

{

Arrays.sort(array);

if (array.length % 2 == 0) {

int mid1 = array.length / 2 - 1;

int mid2 = array.length / 2;

return (array[mid1] + array[mid2]) / 2.0;

} else {

int mid = array.length / 2;

return array[mid];

}

}

pivate static double calculateMode(double[] array) {

Map<Double, Integer>frequencyMap = new HashMap<>();

for (double num : array) {

frequencyMap.put(num, frequencyMap.getOrDefault(num, 0) + 1);

}

double mode = 0;

int maxFrequency = 0;

for (Map.Entry<Double, Integer>entry :frequencyMap.entrySet()) {

if (entry.getValue() >maxFrequency) {

maxFrequency = entry.getValue();

mode = entry.getKey();

}

}

return mode;

}

private static void displayStatistics(double[] array) {

double mean = calculateMean(array);

double median = calculateMedian(array);

double mode = calculateMode(array);

System.out.println("Mean = " + mean);

System.out.println("Median = " + median);

System.out.println("Mode = " + mode);

}

}

15.Write a program to find the number of composite numbers in an array of elements

Sample Input;:

Array of elements = {16, 18, 27, 16, 23, 21, 19}

Sample Output:  
Number of Composite Numbers = 5

Test cases:

1. Array of elements = {26, 28, 37, 26, 33, 31, 29}

2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}

3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}

4. Array of elements = {200, 180, 180, 270, 270, 270, 190, 200}

5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100}

ANSWER:

public class CompositeNumberCounter {

public static void main(String[] args) {

int[] array = {16, 18, 27, 16, 23, 21, 19};

int compositeCount = countCompositeNumbers(array);

System.out.println("Number of Composite Numbers = " + compositeCount);

int[] testArray1 = {26, 28, 37, 26, 33, 31, 29};

int[] testArray2 = {1, 160, 180, 270, 160, 230, 210, 190, 0};

int[] testArray3 = {0, 160, 180, 270, 160, 230, 210, 190, 0};

int[] testArray4 = {200, 180, 180, 270, 270, 270, 190, 200};

int[] testArray5 = {100, 100, 100, 100, 100, 100, 100, 100};

System.out.println("\nTest Case 1:");

displayCompositeCount(testArray1);

System.out.println("\nTest Case 2:");

displayCompositeCount(testArray2);

System.out.println("\nTest Case 3:");

displayCompositeCount(testArray3);

System.out.println("\nTest Case 4:");

displayCompositeCount(testArray4);

System.out.println("\nTest Case 5:");

displayCompositeCount(testArray5);

}

private static int countCompositeNumbers(int[] array) {

int compositeCount = 0;

for (int num : array) {

if (isComposite(num)) {

compositeCount++;

}

}

return compositeCount;

}

private static booleanisComposite(int num) {

if (num< 2) {

return false;

}

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

if (num % i == 0) {

return true;

}

}

return false;

}

private static void displayCompositeCount(int[] array) {

int compositeCount = countCompositeNumbers(array);

System.out.println("Number of Composite Numbers = " + compositeCount);

}

}

16.Write a program to print Right Triangle Star Pattern

Sample Input:: n = 5

Output:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

ANSWER:

import java.util.Scanner;

public class RightTriangleStarPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of rows for the right-angled triangle: ");

int numRows = scanner.nextInt();

printRightTriangle(numRows);

}

private static void printRightTriangle(int numRows) {

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

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

System.out.print(" ");

}

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

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

}

System.out.println();

}

}

}

17,Write a program to print the below pattern?

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | 1 |  |  |  |  |
|  |  |  | 1 |  | 1 |  |  |  |
|  |  | 1 |  | 2 |  | 1 |  |  |
|  | 1 |  | 3 |  | 3 |  | 1 |  |
| 1 |  | 4 |  | 6 |  | 4 |  | 1 |
|  |  |  |  |  |  |  |  |  |

**ANSWER:**

import java.util.Scanner;

public class CharacterPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the character to be printed: ");

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

System.out.print("Max Number of times to be printed: ");

int maxTimes = scanner.nextInt();

printCharacterPattern(characterToPrint, maxTimes);

}

private static void printCharacterPattern(char characterToPrint, int maxTimes) {

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

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

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

}

System.out.println();

}

}

}

18.Write a program to print rectangle symbol pattern.

Get the symbol as input from user

ANSWER:

import java.util.Scanner;

public class RectangleSymbolPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the symbol to create the rectangle pattern: ");

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

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

int numRows = scanner.nextInt();

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

int numCols = scanner.nextInt();

printRectanglePattern(symbol, numRows, numCols);

}

private static void printRectanglePattern(char symbol, int numRows, int numCols) {

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

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

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

}

System.out.println();

}

}

}

1

19.Write a program to print the following pattern

Sample Input:

Enter the number to be printed: 1

Max Number of time printed: 3

1

11

111

11

1

ANSWER:

import java.util.Scanner;

public class PatternPrint {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number to be printed: ");

int num = scanner.nextInt();

System.out.print("Max Number of times printed: ");

int maxTimes = scanner.nextInt();

printPattern(num, maxTimes);

scanner.close();

}

static void printPattern(int num, int maxTimes) {

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

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

System.out.print(num);

}

System.out.println();

}

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

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

System.out.print(num);}

System.out.println();

}

}

}

20.Write a program to print the Inverted Full Pyramid pattern?

ANSWER:

import java.util.Scanner;

public class InvertedFullPyramid {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int numRows = scanner.nextInt();

printInvertedPyramid(numRows);

scanner.close();

}

static void printInvertedPyramid(int numRows) {

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

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

System.out.print(" ");

}

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

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

}

System.out.println();

}

}

}

21.Write a program to print the following pattern

Sample Input:

Enter the Character to be printed: %

Max Number of time printed: 3

%

% %

% % %

ANSWER:

import java.util.Scanner;

public class CharacterPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the character to be printed: ");

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

System.out.print("Max Number of times printed: ");

int maxTimes = scanner.nextInt();

printCharacterPattern(ch, maxTimes);

scanner.close();

}

static void printCharacterPattern(char ch, int maxTimes) {

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

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

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

}

System.out.println();

}

}

}

22.Write a program to print hollow square symbol pattern?

ANSWER:

import java.util.Scanner;

public class HollowSquarePattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int sideLength = scanner.nextInt();

printHollowSquare(sideLength);

scanner.close();

}

static void printHollowSquare(int sideLength) {

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();

}

}

}

23.Write a program to print the below pattern

1

2 2

3 3 3

4 4 4 4

ANSWER:

import java.util.Scanner;

public class NumberTrianglePattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int numRows = scanner.nextInt();

printNumberTriangle(numRows);

scanner.close();

}

static void printNumberTriangle(int numRows) {

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

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

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

}

System.out.println();

}

}

}

24.Write a program to print the below pattern

1

4 9

16 25 36

49 64 81 100

ANSWER:

import java.util.Scanner;

public class SquareNumberPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int numRows = scanner.nextInt();

printSquareNumberPattern(numRows);

scanner.close();

}

static void printSquareNumberPattern(int numRows) {

int currentNumber = 1;

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

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

System.out.print(" ");

}

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

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

currentNumber++;

}

System.out.println();

}

}

}

25.Write a program to print the below pattern

1

2 2

3 3 3

4 4 4 4

3 3 3

2 2

1

ANSWER:

import java.util.Scanner;

public class NumberTriangleMirrorPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int numRows = scanner.nextInt();

printNumberTriangleMirror(numRows);

scanner.close();

}

static void printNumberTriangleMirror(int numRows) {

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

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

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

}

System.out.println();

}

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

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

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

}

System.out.println();

}

}

26.Write a program to print hollow Square Dollar pattern?

ANSWER:

import java.util.Scanner;

public class HollowSquareDollarPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int sideLength = scanner.nextInt();

printHollowSquareDollar(sideLength);

scanner.close();

}

static void printHollowSquareDollar(int sideLength) {

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();

}

}

}

27.Write a program to print inverted pyramid pattern.

Input: no of rows: 3

Output

\*\*\*\*\*

\*\*\*

\*

ANSWER:

import java.util.Scanner;

public class InvertedPyramidPattern {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int numRows = scanner.nextInt();

printInvertedPyramid(numRows);

scanner.close();

}

static void printInvertedPyramid(int numRows) {

int spaces = 0;

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

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

System.out.print(" ");}

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

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

System.out.println();

spaces++;

}

}

}

28.Write a program to reverse a number using loop?(Get the input from user)

Sample Input:

Number: 14567

Sample Output:

Reverse Number: 76541

Test cases:

1. -45721
2. 000
3. AD1947
4. !@#$%
5. 145\*999=144855

ANSWER:

import java.util.Scanner;

public class ReverseNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

if (scanner.hasNextInt()) {

int number = scanner.nextInt();

int reversedNumber = reverseNumber(number);

System.out.println("Reverse Number: " + reversedNumber);

} else {

System.out.println("Invalid input. Please enter a valid integer.");

}

scanner.close();

}

static int reverseNumber(int number) {

int reversedNumber = 0;

while (number != 0) {

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number /= 10;

}

return reversedNumber;

}

}

29.Write a program to convert the given decimal to binary and print the reverse of the binary decimal.

Input: 11

Output: 13

Explanation: (11)10 = (1011)2.

After reversing the bits we get:

(1101)2 = (13)10.

Test cases:

1. 25
2. Eighteen
3. 12
4. -18
5. 34.5

ANSWER:

import java.util.Scanner;

public class ReverseBinary {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

if (scanner.hasNextInt()) {

int decimalNumber = scanner.nextInt();

if (decimalNumber< 0) {

System.out.println("Invalid input. Please enter a non-negative integer.");

} else {

String binaryRepresentation = decimalToBinary(decimalNumber);

System.out.println("Binary Representation: " + binaryRepresentation);

int reversedDecimal = reverseBinary(binaryRepresentation);

System.out.println("Reversed Decimal: " + reversedDecimal);

}

} else {

System.out.println("Invalid input. Please enter a valid integer.");

}

scanner.close();

}

static String decimalToBinary(int decimalNumber) {

return Integer.toBinaryString(decimalNumber);

}

static int reverseBinary(String binaryRepresentation) {

int reversedDecimal = 0;

int length = binaryRepresentation.length();

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

char bit = binaryRepresentation.charAt(i);

if (bit == '1') {

reversedDecimal += Math.pow(2, length - 1 - i);

}

}

return reversedDecimal;

}

}

30.Write a program to find whether the person is eligible for vote or not. And if that particular person is not eligible, then print how many years are left to be eligible.

Sample Input:

Enter your age: 7

Sample output:

You are allowed to vote after 11 years

Test cases:

1. 25
2. Eighteen
3. 12
4. -18
5. 34.5

ANSWER:

import java.util.Scanner;

public class EligibilityForVote {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

if (scanner.hasNextInt()) {

int age = scanner.nextInt();

if (age < 0) {

System.out.println("Invalid age. Please enter a non-negative integer.");

} else {

checkEligibilityForVote(age);

}

} else {

System.out.println("Invalid input. Please enter a valid integer for age.");

}

scanner.close();

}

static void checkEligibilityForVote(int age) {

final int VOTING\_AGE = 18;

if (age >= VOTING\_AGE) {

System.out.println("You are eligible to vote!");

} else {

int yearsLeft = VOTING\_AGE - age;

System.out.println("You are allowed to vote after " + yearsLeft + " years.");

}

}

}

31.Find the LCM and GCD of n numbers?

Sample Input:

N value = 2

Number 1 = 16

Number 2 = 20

Sample Output:

LCM = 80

GCD = 4

Test cases:

1. N = 3, {12, 25, 30}
2. N = 2, {52, 25, 63}
3. N = 3, {17, 19, 11}
4. N = -2, {52, 60}
5. N = 2, {30, 45}

ANSWER:

import java.util.Scanner;

public class LCM\_GCD {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the value of N: ");

int N = scanner.nextInt();

if (N <= 0) {

System.out.println("Invalid input for N. N should be a positive integer.");

return;

}

int[] numbers = new int[N];

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

System.out.print("Enter Number " + (i + 1) + ": ");

numbers[i] = scanner.nextInt();

}

int lcm = calculateLCM(numbers);

int gcd = calculateGCD(numbers);

System.out.println("LCM = " + lcm);

System.out.println("GCD = " + gcd);

}

private static int calculateLCM(int[] numbers) {

int lcm = numbers[0];

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

lcm = (lcm \* numbers[i]) / calculateGCD(lcm, numbers[i]);

}

return lcm;

}

private static int calculateGCD(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

private static int calculateGCD(int[] numbers) {

int gcd = numbers[0];

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

gcd = calculateGCD(gcd, numbers[i]);

}

return gcd;

}

}

32.Write a program using function to calculate the simple interest. Suppose the customer is a senior citizen. He is being offered 12 percent rate of interest; for all other customers, the ROI is 10 percent.

Sample Input:

Enter the principal amount: 200000

Enter the no of years: 3

Is customer senior citizen (y/n): n

Sample Output:

Interest: 60000

Test Cases:

1. Principal: 2000 , Years: 0
2. Principal: 20000 , Years: -2
3. Principal: -2000 , Years: 2
4. Principal: 2 , Years: 2000
5. Principal: 0 , Years: 5

ANSWER:

import java.util.Scanner;

public class SimpleInterestCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the principal amount: ");

double principal = scanner.nextDouble();

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

int years = scanner.nextInt();

System.out.print("Is the customer a senior citizen? (y/n): ");

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

if (principal <= 0 || years < 0) {

System.out.println("Invalid input. Principal amount and number of years should be positive.");

return;

}

double rateOfInterest = (isSeniorCitizen == 'y') ?0.12 : 0.10;

double interest = calculateSimpleInterest(principal, rateOfInterest, years);

System.out.println("Interest: " + interest);

}

private static double calculateSimpleInterest(double principal, double rateOfInterest, int years) {

return principal \* rateOfInterest \* years;

}

}

33.Write a program to print the Fibonacci series.

Sample Input:

Enter the n value: 6

**Sample Output:**

0 1 1 2 3 5

ANSWER:

import java.util.Scanner;

public class FibonacciSeries {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the n value: ");

int n = scanner.nextInt();

if (n <= 0) {

System.out.println("Invalid input. Please enter a positive integer.");

return;

}

System.out.println("Fibonacci Series up to " + n + " terms:");

printFibonacciSeries(n);

}

private static void printFibonacciSeries(int n) {

int firstTerm = 0, secondTerm = 1;

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

System.out.print(firstTerm + "\t");

int nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

secondTerm = nextTerm;

}

}

}

**34:**[Java Program to Find Even Sum of Fibonacci Series Till number N](https://www.geeksforgeeks.org/java-program-to-find-sum-of-fibonacci-series-numbers-of-first-n-even-indexes/)?

Sample Input: n = 4

Sample Output: 33

(N = 4, So here the fibonacci series will be produced from 0th term till 8th term:0, 1, 1, 2, 3, 5, 8, 13, 21

Sum of numbers at even indexes = 0 + 1 + 3 + 8 + 21 = 33)

ANSWER:

import java.util.Scanner;

public class EvenSumFibonacci {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the n value: ");

int n = scanner.nextInt();

if (n <= 0) {

System.out.println("Invalid input. Please enter a positive integer.");

return;

}

System.out.println("Even Sum of Fibonacci Series up to " + n + ": " + findEvenSumFibonacci(n));

}

private static int findEvenSumFibonacci(int n) {

int sum = 0;

int firstTerm = 0, secondTerm = 1;

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

if (i % 2 == 0) {

sum += firstTerm;

}

int nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

secondTerm = nextTerm;

}

return sum;

}

}

35:Write a program to print the numbers from M to N by skipping K numbers in between?

Sample Input:

M = 50

N = 100

K = 7

Sample Output:

50, 58, 66, 74, …..

Test cases:

1. M = 15, N = 05, K = 02
2. .M = 25, N = 50, K = 04
3. M = 15, N = 100, K = -02
4. M = 0 , N = 0 , K = 2
5. M = 200 , N = 200 , K = 50

ANSWER:

import java.util.Scanner;

public class SkipNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the value for M: ");

int M = scanner.nextInt();

System.out.print("Enter the value for N: ");

int N = scanner.nextInt();

System.out.print("Enter the value for K: ");

int K = scanner.nextInt();

if (K <= 0) {

System.out.println("Invalid input. K should be a positive integer.");

return;

}

System.out.println("Numbers from " + M + " to " + N + " by skipping " + K + " numbers:");

printNumbersWithSkip(M, N, K);

}

private static void printNumbersWithSkip(int M, int N, int K) {

for (int i = M; i<= N; i += K) {

System.out.print(i + ", ");

}

}

}

36:Write a program to print all the composite numbers between a and b?

Sample Input:

A = 12

B = 19

Sample Output

14, 15, 16, 18

Test cases:

1. A = 11, B = 11
2. A = 20, B = 10
3. A = 0, B = 0
4. A = -5, B = 5
5. A = 7, B = -12

ANSWER:

import java.util.Scanner;

public class CompositeNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the value for A: ");

int A = scanner.nextInt();

System.out.print("Enter the value for B: ");

int B = scanner.nextInt();

if (A > B) {

System.out.println("Invalid input. A should be less than or equal to B.");

return;

}

System.out.println("Composite numbers between " + A + " and " + B + ":");

printCompositeNumbers(A, B);

}

private static booleanisComposite(int num) {

if (num<= 1) {

return false;

}

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

if (num % i == 0) {

return true;

}

}

return false;

}

private static void printCompositeNumbers(int A, int B) {

for (int i = A; i<= B; i++) {

if (isComposite(i)) {

System.out.print(i + ", ");

}

}

}

}

37:Find the factorial of n?

Sample Input:

N = 4

Sample Output:

4 Factorial = 24

Test cases:

1. N = 0
2. N = -5
3. N = 1
4. N = Q
5. N = 3A

ANSWER:

import java.util.Scanner;

public class Factorial {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the value for N: ");

String input = scanner.next();

if (!input.matches("\\d+")) {

System.out.println("Invalid input. Please enter a non-negative integer.");

return;

}

int N = Integer.parseInt(input);

if (N < 0) {

System.out.println("Invalid input. N should be a non-negative integer.");

return;

}

System.out.println(N + " Factorial = " + calculateFactorial(N));

}

private static long calculateFactorial(int n) {

if (n == 0 || n == 1) {

return 1;

}

long factorial = 1;

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

factorial \*= i;

}

return factorial;

}

}

38.Find the year of the given date is leap year or not

Sample Input:

Enter Date: 04/11/1947

Sample Output:

Given year is Non Leap Year

Test cases:

1. 04/11/19.47
2. 11/15/1936
3. 31/45/1996
4. 64/09/1947
5. 00/00/2000

ANSWER:

import java.util.Scanner;

public class LeapYearChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Date (MM/DD/YYYY): ");

String date = scanner.next();

if (!date.matches("\\d{2}/\\d{2}/\\d{4}")) {

System.out.println("Invalid date format. Please enter date in MM/DD/YYYY format.");

return;

}

int year = Integer.parseInt(date.split("/")[2]);

if (isLeapYear(year)) {

System.out.println("Given year is a Leap Year");

} else {

System.out.println("Given year is a Non Leap Year");

}

}private static booleanisLeapYear(int year) {

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

}

}

39.Find the number of factors for the given number

Sample Input:

Given number: 100

Sample Output:

Number of factors = 9

Test cases:

1. 343
2. 1080
3. -243
4. 101010
5. 0

ANSWER:

import java.util.Scanner;

public class NumberOfFactors {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (number < 0) {

System.out.println("Invalid input. Please enter a non-negative integer.");

return;

}

int factorCount = calculateNumberOfFactors(number);

System.out.println("Number of factors = " + factorCount);

}

private static int calculateNumberOfFactors(int number) {

int count = 0;

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

if (number % i == 0) {

if (number / i == i) {

count++;

} else {

count += 2; // Count both factors (i and number/i)

}

}

}

return count;

}

}

40.Write a program to print the given number is Perfect number or not?

Sample Input:

Given Number: 6

Sample Output:

It’s a Perfect Number

Test cases:

1. 17
2. 26!
3. 143
4. 84.1
5. -963

ANSWER:

import java.util.Scanner;

public class PerfectNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (number <= 0) {

System.out.println("Invalid input. Please enter a positive integer.");

return;

}

if (isPerfectNumber(number)) {

System.out.println("It’s a Perfect Number");

} else {

System.out.println("It's not a Perfect Number");

}

}

private static booleanisPerfectNumber(int number) {

int sum = 1;

public class PerfectNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (number <= 0) {

System.out.println("Invalid input. Please enter a positive integer.");

return;

}

if (isPerfectNumber(number)) {

System.out.println("It’s a Perfect Number");

} else {

System.out.println("It's not a Perfect Number");

}

}

private static booleanisPerfectNumber(int number) {

int sum = 1; // Start with 1 as all numbers are divisible by 1

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

if (number % i == 0) {

sum += i;

if (number / i != i) {

sum += number / i;

}

}

}

return sum == number;

}

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

if (number % i == 0) {

sum += i;

if (number / i != i) {

sum += number / i;

}

}

}

return sum == number;

}

}

41.Write a program to find the square, cube of the given decimal number

Sample Input:

Given Number: 0.6

Sample Output:

Square Number: 0.36

Cube Number:0.216

Test cases:

1. 12
2. 0
3. --0.5
4. 14.25
5. -296

ANSWER:

import java.util.Scanner;

public class SquareAndCube {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

double givenNumber = scanner.nextDouble();

double square = calculateSquare(givenNumber);

double cube = calculateCube(givenNumber);

System.out.println("Square Number: " + square);

System.out.println("Cube Number: " + cube);

}

private static double calculateSquare(double number) {

return Math.pow(number, 2);

}

private static double calculateCube(double number) {

return Math.pow(number, 3);

}

}

42.Find the nth odd number after n odd number

Sample Input: N : 7

Sample Output:

Hence the values printed for i are 1 , 3 , 5.

Test cases:

1. N = 0
2. N = -6
3. N = 2021
4. N = -14.5
5. N = -196

ANSWER:

import java.util.Scanner;

public class NthOddNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the value for N: ");

int N = scanner.nextInt();

if (N < 0) {

System.out.println("Invalid input. N should be a non-negative integer.");

return;

}

int oddNumber = findNthOddNumber(N);

System.out.println("The nth odd number after " + N + " odd numbers is: " + oddNumber);

}

private static int findNthOddNumber(int N) {

int count = 0;

int number = 1;

while (count < N) {

number += 2;

count++;

}

return number;

}

}

43.Program to find whether the given number is Armstrong number or not

Sample Input:

Enter number: 153

Sample Output:

Given number is Armstrong number

Test cases:

1. 370
2. 1
3. 371
4. 145678
5. 0.21345

ANSWER:

import java.util.Scanner;

public class ArmstrongNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (number < 0) {

System.out.println("Invalid input. Please enter a non-negative integer.");

return;

}

if (isArmstrongNumber(number)) {

System.out.println("Given number is Armstrong number");

} else {

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

}

}

private static booleanisArmstrongNumber(int number) {

int originalNumber = number;

int sum = 0;

int digits = countDigits(number);

while (number > 0) {

int digit = number % 10;

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

number /= 10;

}

return sum == originalNumber;

}

private static int countDigits(int number) {

int count = 0;

while (number > 0) {

number /= 10;

count++;

}

return count;

}

}

44.Write a program to find the sum of digits of N digit number (sum should be single digit)

Sample Input:

Enter N value: 3

Enter 3 digit numbers: 143

Test cases:

1. N = 2, 158
2. N = 3, 14
3. N = 4, 0148
4. N = 1, 0004
5. N = 4, 7263

ANSWER:

import java.util.Scanner;

public class SingleDigitSum {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter N value: ");

int N = scanner.nextInt();

if (N <= 0) {

System.out.println("Invalid input. N should be a positive integer.");

return;

}

System.out.print("Enter " + N + " digit number: ");

int num = scanner.nextInt();

if (countDigits(num) != N) {

System.out.println("Invalid input. Please enter a " + N + " digit number.");

return;

}

int singleDigitSum = calculateSingleDigitSum(num);

System.out.println("Sum of digits (single digit): " + singleDigitSum);

}

private static int countDigits(int number) {

int count = 0;

while (number > 0) {

number /= 10;

count++;

}

return count;

}

private static int calculateSingleDigitSum(int number) {

int sum = 0;

while (number > 0) {

sum += number % 10;

number /= 10;

}

while (sum >= 10) {

sum = sum % 10 + sum / 10;

}

return sum;

}

}

45.Write a program to find the square root of a perfect square number(print both the positive and negative values)

Sample Input:

Enter the number: 6561

Sample Output:

Square Root: 81, -81

Test cases:

1. 1225
2. 9801
3. 1827
4. -100
5. 0

ANSWER:

import java.util.Scanner;

public class SquareRoot {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

if (number < 0) {

System.out.println("Invalid input. Please enter a non-negative integer.");

return;

}

double squareRoot1 = Math.sqrt(number);

double squareRoot2 = -Math.sqrt(number);

System.out.println("Square Root: " + squareRoot1 + ", " + squareRoot2);

}

}

**46.Write a program to given an integer n, return true if it is a power of three. Otherwise, return false.**

**Input =27**

**Output= true**

**Explanation: 27=33**

**Test cases:**

1. **12**
2. **abc@45**
3. **1827**
4. **-100**
5. **0**

**ANSWER;**

public class PowerOfThree {

public static void main(String[] args) {

System.out.println(isPowerOfThree(12));

System.out.println(isPowerOfThree("abc@45"));

System.out.println(isPowerOfThree(1827));

System.out.println(isPowerOfThree(-100));

System.out.println(isPowerOfThree(0));

System.out.println(isPowerOfThree(27));

}

public static booleanisPowerOfThree(int n) {

if (n <= 0) {

return false;

}

while (n % 3 == 0) {

n /= 3;

}

return n == 1;

}

public static booleanisPowerOfThree(String input) {

try {

int n = Integer.parseInt(input);

return isPowerOfThree(n);

} catch (NumberFormatException e) {

return false;

}

}

}

**47.Write a program to given a string paragraph and a string array of the banned words banned, return the most frequent word that is not banned. It is guaranteed there is at least one word that is not banned, and that the answer is unique.**

**Input Paragraph=”Ram hit a ball, the hit ball flew far after it was hit”,**

**Banned = [hit]**

**Output=”Ball”**

**ANSWER:**

import java.util.HashMap;

import java.util.HashSet;

import java.util.Map;

public class MostFrequentWord {

public static void main(String[] args) {

String paragraph = "Ram hit a ball, the hit ball flew far after it was hit";

String[] banned = {"hit"};

String result = mostCommonWord(paragraph, banned);

System.out.println("Output: " + result); // Output: "ball"

}

public static String mostCommonWord(String paragraph, String[] banned) {

paragraph = paragraph.toLowerCase().replaceAll("[^a-z ]", " "); // Convert to lowercase and remove non-alphabetic characters

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

HashSet<String>bannedSet = new HashSet<>();

for (String word : banned) {

bannedSet.add(word);

}

Map<String, Integer>frequencyMap = new HashMap<>();

String mostFrequentWord = "";

int maxFrequency = 0;

for (String word : words) {

if (!bannedSet.contains(word) && !word.isEmpty()) {

int frequency = frequencyMap.getOrDefault(word, 0) + 1;

frequencyMap.put(word, frequency);

if (frequency >maxFrequency) {

maxFrequency = frequency;

mostFrequentWord = word;

}

}

}

return mostFrequentWord;

}

}

**48.Write a program to given a fixed-length integer array arr, duplicate each occurrence of zero, shifting the remaining elements to the right.**

**Input: arr = [1, 0, 2, 3, 0, 4, 5, 0]**

**Output: [1, 0, 0, 2, 3, 0, 0, 4]**

**Explanation: After calling your function, the input array is modified to [1, 0, 0, 2, 3, 0, 0, 4]**

**ANSWER;**

public class DuplicateZeros {

public static void main(String[ ] args) {

int[ ] arr = {1, 0, 2, 3, 0, 4, 5, 0};

duplicateZeros(arr);

System.out.print("Output: ");

for (int num :arr) {

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

}

}

public static void duplicateZeros(int[ ] arr) {

int length = arr.length;

int zeroCount = 0;

for (int num :arr) {

if (num == 0) {

zeroCount++;

}

}

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

if (i + zeroCount< length) {

arr[i + zeroCount] = arr[i];

}

if (arr[i] == 0) {

zeroCount--;

if (i + zeroCount< length) {

arr[i + zeroCount] = 0;

}

}

}

}

}

**49.Write a program to given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array.**

**Input nums = [3, 0, 1]**

**Output: 2**

**Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0, 3]. 2 is the missing number in the range since it does not appear in nums.**

**ANSWER:**

public class MissingNumber {

public static void main(String[] args) {

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

int result = findMissingNumber(nums);

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

}

public static int findMissingNumber(int[] nums) {

int n = nums.length;

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

int actualSum = 0;

for (int num :nums) {

actualSum += num;

}

return expectedSum - actualSum;

}

}

**50.Write a program to given an integer array nums, find the subarray with the largest sum, and return its sum.**

**Input nums = [-2, 1,-3, 4,-1, 2, 1,-5, 4]**

**Output: 6**

**Explanation: The subarray [4,-1, 2, 1] has the largest sum 6.**

**ANSWER:**

public class MaxSubarraySum {

public static int maxSubArray(int[] nums) {

int maxSum = nums[0];

int currentSum = nums[0];

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

currentSum = Math.max(nums[i], currentSum + nums[i]);

maxSum = Math.max(maxSum, currentSum);

}

return maxSum;

}

public static void main(String[] args) {

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

int result = maxSubArray(nums);

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

}

}

**51.Write a java program to print the multiplication table of number m up to n.**

**Sample Input:**

**M = 4**

**N = 5**

**Sample Output:**

**1x4=4**

**2x4=8**

**3x4=12**

**4x4=16**

**5x4=20**

**Test cases:**

**M = 6, N = -3**

**M = -3, N = 5**

**M = 4, N = 0**

**M = 0, N = 0**

**M = -5, N = -5**

**ANSWER:**

import java.util.Scanner;

public class MultiplicationTable

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the value of M: ");

int m = scanner.nextInt();

System.out.print("Enter the value of N: ");

int n = scanner.nextInt();

if (n < 0)

{

System.out.println("Invalid input. N should be a non-negative integer.");

return;

}

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

{

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

}

}

}

1. **Write Java programs to implement multiple threads and apply join method for thread and thread has to be started after 500ms using sleep ().**

**ANSWER:**

class MyThread extends Thread {

public void run()

{

try

{

Thread.sleep(500);

System.out.println(Thread.currentThread().getName() + " is running.");

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

}

public class JoinExampleThread {

public static void main(String[] args) {

MyThread t1 = new MyThread();

MyThread t2 = new MyThread();

t1.start();

t2.start();

try

{

t1.join();

t2.join();

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println("Main thread is finished.");

}

}

**54.Create a simple generics class with type parameters for sorting values of different types.**

**ANSWER:**

import java.util.Arrays;

public class GenericSorter<T extends Comparable<T>> {

private T[ ] array;

public GenericSorter(T[ ] array) {

this.array = array;

}

public void sort( ) {

Arrays.sort(array);

}

public void displaySortedArray() {

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

}

public static void main(String[ ] args) {

Integer[ ] intArray = {5, 2, 8, 1, 3};

GenericSorter<Integer>intSorter = new GenericSorter<>(intArray);

intSorter.sort();

intSorter.displaySortedArray();

String[] stringArray = {"apple", "orange", "banana", "grape", "kiwi"};

GenericSorter<String>stringSorter = new GenericSorter<>(stringArray);

stringSorter.sort();

stringSorter.displaySortedArray();

}

}

**55.Create a class name ‘overload’. Write a program to assign the values for two values by different number of arguments using a single function.**

**ANSWER:**

public class Overload {

public void assignValues(int a, int b) {

System.out.println("Assigning values using two arguments:");

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

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

}

public void assignValues(int a, int b, int c) {

System.out.println("Assigning values using three arguments:");

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

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

System.out.println("Value of c: " + c);

}

public void assignValues(int a, int b, int c, int d) {

System.out.println("Assigning values using four arguments:");

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

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

System.out.println("Value of c: " + c);

System.out.println("Value of d: " + d);

}

public static void main(String[] args)

{

Overload overloadObj = new Overload();

overloadObj.assignValues(10, 20);

overloadObj.assignValues(30, 40, 50);

overloadObj.assignValues(60, 70, 80, 90);

}

}

1. **Write a Java Program to count the number of words in a string using Hash Map.**

**ANSWER:**

import java.util.HashMap;

import java.util.Map;

public class WordCount

{

public static void main(String[ ] args)

{

String inputString = "This is a sample string. This string has words, and we want to count them.";

Map<String, Integer>wordFrequencyMap = new HashMap<>();

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

for (String word : words)

{

word = word.replaceAll("[^a-zA-Z]", "").toLowerCase();

wordFrequencyMap.put(word, wordFrequencyMap.getOrDefault(word, 0) + 1);

}

System.out.println("Word Count:");

for (Map.Entry<String, Integer>entry :wordFrequencyMap.entrySet())

{

System.out.println(entry.getKey() + ": " + entry.getValue());

}

}

}

1. **Write a Java Program to read an email and password from excel sheet by retrieving the cell using getRow() and getCell() method.**

**ANSWERS;**

import org.apache.poi.ss.usermodel.\*;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.FileInputStream;

import java.io.IOException;

public class ExcelReader {

public static void main(String[] args) {

String excelFilePath = "path/to/your/excel/file.xlsx";

try (FileInputStreaminputStream = new FileInputStream(excelFilePath);

Workbook workbook = new XSSFWorkbook(inputStream)) {

Sheet sheet = workbook.getSheetAt(0);

Row emailRow = sheet.getRow(0);

Cell emailCell = emailRow.getCell(0);

String email = emailCell.getStringCellValue();

Row passwordRow = sheet.getRow(1);

Cell passwordCell = passwordRow.getCell(0);

String password = passwordCell.getStringCellValue();

System.out.println("Email: " + email);

System.out.println("Password: " + password);

}

catch (IOException e)

{

e.printStackTrace();

}

}

}

1. **Write a Java program to sorts the given value using Hash Map.**

**ANSWER;**

import java.util.HashMap;

import java.util.Map;

import java.util.TreeMap;

public class SortHashMap {

public static void main(String[ ] args) {

Map<String, Integer>unsortedMap = new HashMap<>();

unsortedMap.put("Alice", 30);

unsortedMap.put("Bob", 25);

unsortedMap.put("Charlie", 35);

unsortedMap.put("David", 28);

TreeMap<String, Integer>sortedMap = new TreeMap<>(unsortedMap);

System.out.println("Unsorted HashMap:");

displayMap(unsortedMap);

System.out.println("\nSortedTreeMap:");

displayMap(sortedMap);

}

private static void displayMap(Map<String, Integer> map) {

for (Map.Entry<String, Integer>entry :map.entrySet()) {

System.out.println(entry.getKey() + ": " + entry.getValue());

}

}

}

**59.Write a Java program to find distinct characters and their count in a string.**

**ANSWER:**

import java.util.HashMap;

import java.util.Map;

public class DistinctCharactersCount {

public static void main(String[] args) {

String inputString = "programming";

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

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

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

}

System.out.println("Distinct Characters and their Count:");

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

System.out.println(entry.getKey() + ": " + entry.getValue());

}

}

}

**60.Write a program to print all the unique characters in a String. For instance, if the input string is “abcb”, the output will be the characters ‘a’ and ‘c’ as they are unique. The character ‘b’ repeats twice and so it will not be printed.**

**ANSWER:**

import java.util.HashSet;

import java.util.Set;

public class UniqueCharacters {

public static void main(String[] args) {

String inputString = "abcb";

Set<Character>uniqueCharacters = new HashSet<>();

Set<Character>repeatedCharacters = new HashSet<>();

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

set

if (!uniqueCharacters.add(ch)) {

repeatedCharacters.add(ch);

}

}

uniqueCharacters.removeAll(repeatedCharacters);

System.out.println("Unique Characters in the String:");

for (char uniqueChar :uniqueCharacters) {

System.out.println(uniqueChar);

}

}

}