EASY:

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

Sample Input:

String: TEMPLE

Sample Output:

Reverse String: ELPMET

Test cases:

1. SIGN UP
2. AT-LEAST
3. 1245
4. !@#$%
5. 145\*999=144855

Code:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String input = scanner.nextLine();

String reversedString = reverseWord(input);

System.out.println("Reverse String: " + reversedString);

// Test cases

String[] testCases = {"SIGN UP", "AT-LEAST", "1245", "!@#$%", "145\*999=144855"};

for (String testCase : testCases) {

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

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

}

}

public static String reverseWord(String input) {

StringBuilder reversed = new StringBuilder();

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

reversed.append(input.charAt(i));

}

return reversed.toString();

}

}

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

Sample Input:

Enter the user name: Saveetha@789

Reenter the user name: Saveetha@123

Sample Output:

User name is Invalid

Code:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the user name: ");

String username1 = scanner.nextLine();

System.out.print("Reenter the user name: ");

String username2 = scanner.nextLine();

if (isValidUsername(username1, username2)) {

System.out.println("User name is Valid");

} else {

System.out.println("User name is Invalid");

}

}

public static boolean isValidUsername(String username1, String username2) {

return username1.equals(username2);

}

}

1. 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

Code:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String input = scanner.nextLine();

// Check if input contains only digits

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

long number = Long.parseLong(input);

long reversedNumber = reverseNumber(number);

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

} else {

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

}

}

public static long reverseNumber(long number) {

long reversed = 0;

while (number != 0) {

long digit = number % 10;

reversed = reversed \* 10 + digit;

number /= 10;

}

return reversed;

}

}

1. 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

Code:

import java.util.Scanner;

public class Main {

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 >= 18) {

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

} else {

int yearsLeft = 18 - age;

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

}

} else {

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

}

}

}

1. 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}

Code:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

if (n < 1) {

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

return;

}

int[] numbers = new int[n];

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

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

numbers[i] = scanner.nextInt();

}

int lcm = findLCM(numbers);

int gcd = findGCD(numbers);

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

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

}

public static int findLCM(int[] numbers) {

int lcm = numbers[0];

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

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

}

return lcm;

}

public static int findGCD(int[] numbers) {

int gcd = numbers[0];

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

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

}

return gcd;

}

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

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

}

1. Write a program to print Right Triangle Star Pattern

Sample Input:: n = 5

Output:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

Code:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

// Print right triangle star pattern

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

// Print spaces

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

System.out.print(" "); // Two spaces for formatting

}

// Print stars

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

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

}

// Move to the next line

System.out.println();

}

}

}

1. Write a program to print the below pattern?

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

Code:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

// Print the pattern

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

// Print spaces

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

System.out.print(" "); // Two spaces for formatting

}

// Print numbers before the middle number

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

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

}

// Print numbers after the middle number

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

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

}

// Move to the next line

System.out.println();

}

}

// Function to calculate the number at a given row and column in Pascal's triangle

public static int getNumber(int row, int col) {

if (col == 0 || col == row) {

return 1;

} else {

return getNumber(row - 1, col - 1) + getNumber(row - 1, col);

}

}

}

1. 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

CODE:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Get input from the user

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 customer a senior citizen (y/n): ");

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

// Calculate the simple interest

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

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

// Display the result

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

}

// Function to calculate simple interest

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

// Validate input

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

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

return 0;

}

// Calculate interest

return principal \* rateOfInterest \* years;

}

}

1. [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)

CODE:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int evenSum = evenFibonacciSum(n);

System.out.println("Sum of even numbers in Fibonacci series till " + n + " terms: " + evenSum);

}

public static int evenFibonacciSum(int n) {

int sum = 0;

int a = 0;

int b = 1;

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

if (i % 2 == 0) {

sum += a;

}

int nextTerm = a + b;

a = b;

b = nextTerm;

}

return sum;

}

}

1. 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

CODE:

import java.util.Scanner;

public class Main {

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();

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

int k = scanner.nextInt();

printNumbers(m, n, k);

}

public static void printNumbers(int m, int n, int k) {

if (k <= 0) {

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

return;

}

if (m > n) {

System.out.println("Invalid input! M should be less than or equal to N.");

return;

}

for (int i = m; i <= n; i += k) {

System.out.print(i);

if (i + k <= n) {

System.out.print(", ");

}

}

}

}

1. Write a program for matrix addition?

Sample Input:

Mat1 = 1 2

5 3

Mat2 = 2 3

4 1

Sample Output:

Mat Sum = 3 5

1. 4

CODE:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of rows and columns for the matrices:");

int rows = scanner.nextInt();

int columns = scanner.nextInt();

// Input matrices

int[][] mat1 = new int[rows][columns];

int[][] mat2 = new int[rows][columns];

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

inputMatrix(scanner, mat1);

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

inputMatrix(scanner, mat2);

// Perform matrix addition

int[][] matSum = addMatrices(mat1, mat2);

// Display the result

System.out.println("Matrix Sum:");

displayMatrix(matSum);

}

// Function to input matrix elements

public static void inputMatrix(Scanner scanner, int[][] matrix) {

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

for (int j = 0; j < matrix[0].length; j++) {

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

}

}

}

// Function to add two matrices

public static int[][] addMatrices(int[][] mat1, int[][] mat2) {

int rows = mat1.length;

int columns = mat1[0].length;

int[][] matSum = new int[rows][columns];

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

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

matSum[i][j] = mat1[i][j] + mat2[i][j];

}

}

return matSum;

}

// Function to display matrix elements

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

for (int[] row : matrix) {

for (int element : row) {

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

}

System.out.println();

}

}

}

1. Write a program to print rectangle symbol pattern.

Get the symbol as input from user

CODE:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the symbol you want to use: ");

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

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

int rows = scanner.nextInt();

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

int columns = scanner.nextInt();

// Print rectangle symbol pattern

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

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

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

}

System.out.println();

}

}

}

1. 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

CODE:

import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

// Get names from the user

System.out.println("Enter names (enter 'done' to finish):");

while (true) {

String name = scanner.nextLine();

if (name.equals("done")) {

break;

}

names.add(name);

}

// Get sorting order from the user

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

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

// Sort names based on the user's choice

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

Collections.sort(names);

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

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

} else {

System.out.println("Invalid input! Please enter 'A' or 'D'.");

return;

}

// Print the sorted names

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

for (String name : names) {

System.out.println(name);

}

}

}

1. Write a program for matrix multiplication?

Sample Input:

Mat1 = 1 2

5 3

Mat2 = 2 3

4 1

Sample Output:

Mat Sum = 10 5

1. 18

CODE:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of rows and columns for the first matrix:");

int rows1 = scanner.nextInt();

int columns1 = scanner.nextInt();

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

int[][] mat1 = inputMatrix(scanner, rows1, columns1);

System.out.println("Enter the number of rows and columns for the second matrix:");

int rows2 = scanner.nextInt();

int columns2 = scanner.nextInt();

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

int[][] mat2 = inputMatrix(scanner, rows2, columns2);

if (columns1 != rows2) {

System.out.println("Matrix multiplication is not possible!");

return;

}

int[][] product = multiplyMatrices(mat1, mat2);

System.out.println("Matrix Product:");

displayMatrix(product);

}

public static int[][] inputMatrix(Scanner scanner, int rows, int columns) {

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

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

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

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

}

}

return matrix;

}

public static int[][] multiplyMatrices(int[][] mat1, int[][] mat2) {

int rows1 = mat1.length;

int columns1 = mat1[0].length;

int columns2 = mat2[0].length;

int[][] product = new int[rows1][columns2];

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < columns2; j++) {

for (int k = 0; k < columns1; k++) {

product[i][j] += mat1[i][k] \* mat2[k][j];

}

}

}

return product;

}

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

for (int[] row : matrix) {

for (int element : row) {

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

}

System.out.println();

}

}

}

1. 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

CODE:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

System.out.print("Enter the maximum number of times printed: ");

int maxTimes = scanner.nextInt();

printPattern(number, maxTimes);

}

public static void printPattern(int number, int maxTimes) {

// Print pattern in ascending order

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

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

System.out.print(number);

}

System.out.println();

}

// Print pattern in descending order

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

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

System.out.print(number);

}

System.out.println();

}

}

}

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

CODE:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String line = scanner.nextLine();

int specialCharacterCount = printSpecialCharacters(line);

System.out.println("Number of special characters in the line: " + specialCharacterCount);

}

public static int printSpecialCharacters(String line) {

int count = 0;

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

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

char ch = line.charAt(i);

if (!Character.isLetterOrDigit(ch) && !Character.isWhitespace(ch)) {

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

count++;

}

}

System.out.println(); // Print a new line for better formatting

return count;

}

}

1. 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

CODE:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int a = scanner.nextInt();

System.out.print("Enter 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);

}

public static void printCompositeNumbers(int a, int b) {

for (int i = a; i <= b; i++) {

if (isComposite(i)) {

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

}

}

System.out.println(); // Print a new line for better formatting

}

public static boolean isComposite(int number) {

if (number <= 1) {

return false;

}

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

if (number % i == 0) {

return true;

}

}

return false;

}

}

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

CODE:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int rows = scanner.nextInt();

// Print inverted full pyramid pattern

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

// Print spaces

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

System.out.print(" ");

}

// Print asterisks

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

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

}

// Move to the next line

System.out.println();

}

}

}

1. 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}

CODE:

import java.util.Arrays;

import java.util.HashMap;

import java.util.Map;

public class Main {

public static void main(String[] args) {

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

double mean = findMean(numbers);

double median = findMedian(numbers);

double mode = findMode(numbers);

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

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

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

}

public static double findMean(double[] numbers) {

double sum = 0;

for (double num : numbers) {

sum += num;

}

return sum / numbers.length;

}

public static double findMedian(double[] numbers) {

Arrays.sort(numbers);

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

int mid1 = numbers.length / 2 - 1;

int mid2 = numbers.length / 2;

return (numbers[mid1] + numbers[mid2]) / 2;

} else {

return numbers[numbers.length / 2];

}

}

public static double findMode(double[] numbers) {

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

for (double num : numbers) {

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) {

mode = entry.getKey();

maxFrequency = entry.getValue();

}

}

return mode;

}

}

1. 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

CODE:

import java.util.Scanner;

public class Main {

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("Factorial is not defined for negative numbers.");

} else {

long factorial = findFactorial(n);

System.out.println(n + " Factorial = " + factorial);

}

}

public static long findFactorial(int n) {

if (n == 0) {

return 1;

} else {

long factorial = 1;

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

factorial \*= i;

}

return factorial;

}

}

}