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

**Program:**

**import java.util.\*;**

**class revstr{**

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

**Scanner input=new Scanner(System.in);**

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

**String n=input.nextLine();**

**String revstr="";**

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

**revstr=n.charAt(i)+revstr;**

**}**

**System.out.print("Reverse String: "+revstr);**

**}**

**}**

1. Write a program to check whether the entered user name is valid. 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

**Program:**

**import java.util.\*;**

**class username{**

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

**Scanner input=new Scanner(System.in);**

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

**String a=input.nextLine();**

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

**String b=input.nextLine();**

**if(a==b){**

**System.out.print("User name is valid");**

**}**

**else{**

**System.out.print("User name is Invalid");**

**}**

**}**

**}**

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
6. 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

**Program**:

**import java.util.\*;**

**class vote{**

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

**Scanner input=new Scanner(System.in);**

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

**int n=input.nextInt();**

**int a;**

**if(n>=18){**

**System.out.print("You are eligible to vote");**

**}**

**else if(n<18 && n>0){**

**a=18-n;**

**System.out.print("Your are eligible to vote after "+a+" years");**

**}**

**else{**

**System.out.print("Invalid 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}

**Program:**

**import java.util.\*;**

**class lcmgcd{**

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

**int arr[]={16,20};**

**int lcm=arr[0];**

**int gcd=arr[0];**

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

**gcd=findGCD(arr[i],lcm);**

**lcm=(lcm\*arr[i])/gcd;**

**}**

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

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

**}**

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

**if(b==0)**

**return a;**

**return findGCD(b,a%b);**

**}**

**}**

1. Write a program to print Right Triangle Star Pattern

Sample Input:: n = 5

Output:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

1. Write a program to print the below pattern?

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

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

**Program:**

**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("Are you a senior citizen? (yes/no): ");**

**String seniorCitizenStatus = scanner.next();**

**double rateOfInterest;**

**if (seniorCitizenStatus.equalsIgnoreCase("yes")) {**

**rateOfInterest = 0.12;**

**} else {**

**rateOfInterest = 0.10;**

**}**

**double simpleInterest = calculateSimpleInterest(principal, rateOfInterest, years);**

**System.out.println("Simple Interest: " + simpleInterest);**

**scanner.close();**

**}**

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

**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)

**Program:**

**import java.util.Scanner;**

**public class fibo{**

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

**Scanner input=new Scanner(System.in);**

**System.out.print("enter a number:");**

**int n=input.nextInt();**

**int i,n1=0,n2=1,n3;**

**System.out.print(n1+","+n2);**

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

**n3=n1+n2;**

**System.out.print(","+n3);**

**n1=n2;**

**n2=n3;**

**}**

**}**

**}**

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

**Program:**

**import java.util.Scanner;**

**public class PrintNumbersWithSkip {**

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

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

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

**int m = scanner.nextInt();**

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

**int n = scanner.nextInt();**

**System.out.print("Enter the number to skip (K): ");**

**int k = scanner.nextInt();**

**if (m > n) {**

**System.out.println("Starting number (M) should be less than or equal to ending number (N).");**

**return;**

**}**

**System.out.println("Numbers from " + m + " to " + n + " with a skip of " + k + ":");**

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

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

**}**

**scanner.close();**

**}**

**}**

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]

**Program:**

**class MatrixAddition{**

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

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

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

**int c[][]=new int[2][2];**

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

**for(int j=0;j<2;j++){**

**c[i][j]=a[i][j]+b[i][j];**

**System.out.print(c[i][j]+" ");**

**}**

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

**}**

**}}**

1. Write a program to print rectangle symbol pattern.

Get the symbol as input from user

**Program:**

**import java.util.Scanner;**

**public class RectanglePattern {**

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

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

**System.out.print("Enter a symbol: ");**

**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();**

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

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

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

**}**

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

**}**

**scanner.close();**

**}**

**}**

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

**Program:**

**import java.util.ArrayList;**

**import java.util.Collections;**

**import java.util.List;**

**import java.util.Scanner;**

**public class NameSorter {**

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

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

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

**while (true) {**

**System.out.print("Enter a name (or 'stop' to finish): ");**

**String name = scanner.nextLine();**

**if (name.equalsIgnoreCase("stop")) {**

**break;**

**}**

**names.add(name);**

**}**

**System.out.print("Enter 'asc' for ascending or 'desc' for descending order: ");**

**String sortOrder = scanner.nextLine();**

**if (sortOrder.equalsIgnoreCase("asc")) {**

**Collections.sort(names);**

**} else if (sortOrder.equalsIgnoreCase("desc")) {**

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

**} else {**

**System.out.println("Invalid sorting order specified. Names will not be sorted.");**

**}**

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

**for (String name : names) {**

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

**}**

**scanner.close();**

**}**

**}**

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]

**Program:**

**public class MatrixMultiplicationExample{**

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

**int a[][]={{1,2},{5,3}};**

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

**int c[][]=new int[2][2];**

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

**for(int j=0;j<2;j++){**

**c[i][j]=0;**

**for(int k=0;k<2;k++)**

**{**

**c[i][j]+=a[i][k]\*b[k][j];**

**}**

**System.out.print(c[i][j]+" ");**

**}**

**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

**Program:**

**import vava.util.Scanner;**

**public class Pattern {**

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

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

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

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

**System.out.println("Max Number of time printed: ");**

**int maxNumber = scanner.nextInt();**

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

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

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

**}**

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

**}**

**for (int i = maxNumber - 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?

**Program**:

**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();**

**int specialCharacterCount = 0;**

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

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

**char ch = inputLine.charAt(i);**

**if (!Character.isLetterOrDigit(ch)) {**

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

**specialCharacterCount++;**

**}**

**}**

**System.out.println("Number of special characters: " + specialCharacterCount);**

**scanner.close();**

**}**

**}**

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

**Program:**

**import java.util.Scanner;**

**public class CompositeNumberFinder {**

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

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

**System.out.print("Enter start and end of the range (a b): ");**

**int a = scanner.nextInt();**

**int b = scanner.nextInt();**

**System.out.println("Composite numbers between " + a + " and " + b + ":");**

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

**if (i > 1 && isComposite(i)) {**

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

**}**

**}**

**scanner.close();**

**}**

**static boolean isComposite(int n) {**

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

**if (n % i == 0) {**

**return true;**

**}**

**}**

**return false;**

**}**

**}**

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

**Program:**

**import java.util.Scanner;**

**public class InvertedFullPyramid {**

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

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

**// Get the number of rows**

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

**int rows = scanner.nextInt();**

**// Print the inverted full pyramid**

**for (int i = rows; i >= 1; i--) {**

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

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

**}**

**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}

**Program:**

**import java.util.\*;**

**public class MeanMedianMode {**

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

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

**System.out.println("Enter the array of numbers: ");**

**String[] numbersString = scanner.nextLine().split(" ");**

**int[] numbers = new int[numbersString.length];**

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

**numbers[i] = Integer.parseInt(numbersString[i]);**

**}**

**double mean = 0;**

**for (int number : numbers) {**

**mean += number;**

**}**

**mean /= numbers.length;**

**Arrays.sort(numbers);**

**int medianIndex = numbers.length / 2;**

**double median = numbers[medianIndex];**

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

**median = (median + numbers[medianIndex - 1]) / 2;**

**}**

**int[] modes = new int[numbers.length];**

**int maxCount = 0;**

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

**int count = 0;**

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

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

**count++;**

**}**

**}**

**if (count > maxCount) {**

**maxCount = count;**

**modes = new int[1];**

**modes[0] = numbers[i];**

**} else if (count == maxCount) {**

**modes = Arrays.copyOf(modes, modes.length + 1);**

**modes[modes.length - 1] = numbers[i];**

**}**

**}**

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

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

**System.out.println("Mode = " + Arrays.toString(modes));**

**}**

**}**

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

**Program:**

**import java.util.Scanner;**

**public class Factorial {**

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

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

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

**if (scanner.hasNextInt()) {**

**int N = scanner.nextInt();**

**if (N == 0 || N == 1) {**

**System.out.println(N + " Factorial = " + 1);**

**} else if (N >= 0) {**

**int factorial = 1;**

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

**factorial \*= i;**

**}**

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

**} else {**

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

**}**

**} else {**

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

**}**

**scanner.close();**

**}**

**}**

1. Write a program to print the following pattern

Sample Input:

Enter the Character to be printed: %

Max Number of time printed: 3

%

% %

% % %

**Program:**

**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 character = scanner.next().charAt(0);**

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

**int maxTimes = scanner.nextInt();**

**scanner.close();**

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

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

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

**}**

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

**}**

**}**

**}**

1. 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

**Program:**

**import java.time.LocalDate;**

**import java.time.format.DateTimeFormatter;**

**import java.util.Scanner;**

**public class LeapYear {**

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

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

**System.out.println("Enter Date: ");**

**String dateString = scanner.nextLine();**

**DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd/MM/yyyy");**

**LocalDate date = LocalDate.parse(dateString, formatter);**

**if (date.isLeapYear()) {**

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

**} else {**

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

**}**

**}**

**}**

1. 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

**Program:**

**import java.util.Scanner;**

**public class NumberOfFactors {**

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

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

**System.out.println("Enter the number: ");**

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

**int count = 0;**

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

**if (number % i == 0) {**

**count++;**

**if (i != number / i) {**

**count++;**

**}**

**}**

**}**

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

**}**

**}**

1. 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

**Program:**

**import java.util.Scanner;**

**public class PerfectNumber {**

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

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

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

**if (scanner.hasNextInt()) {**

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

**if (number > 0) {**

**int sum = 1;**

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

**if (number % i == 0) {**

**sum += i;**

**if (i != number / i) {**

**sum += number / i;**

**}**

**}**

**}**

**if (sum==number) {**

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

**} else {**

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

**}**

**} else {**

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

**}**

**} else {**

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

**}**

**scanner.close();**

**}**

**}**

1. 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

Test cases:

1. India is my country
2. All are my brothers and sisters
3. Why dry sky
4. Shy Try Cry
5. EDUCATION

**Program:**

**import java.util.Scanner;**

**public class CountingVowels {**

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

**int count = 0;**

**System.out.println("Enter a sentence :");**

**Scanner sc = new Scanner(System.in);**

**String sentence = sc.nextLine();**

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

**char ch = sentence.charAt(i);**

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

**count ++;**

**}**

**}**

**System.out.println("Number of vowels in the given sentence is "+count);**

**}**

**}**

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

Get the symbol from user.

**Program:**

**import java.util.Scanner;**

**public class HollowSquareSymbolPattern {**

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

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

**System.out.println("Enter the symbol: ");**

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

**System.out.println("Enter the size of the square: ");**

**int size = scanner.nextInt();**

**// Print the top side of the square**

**for (int i = 0; i < size; i++) {**

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

**}**

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

**// Print the middle of the square**

**for (int i = 0; i < size - 2; i++) {**

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

**for (int j = 0; j < size - 2; j++) {**

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

**}**

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

**}**

**// Print the bottom side of the square**

**for (int i = 0; i < size; i++) {**

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

**}**

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

**}**

**}**

1. 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

Test cases:

1. TRY
2. MEDIAN
3. ONE
4. KNOWLEDGE
5. EDUCATION

**Program:**

**import java.util.Scanner;**

**public class ConsonantsAndVowelsSeparator {**

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

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

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

**String word = scanner.nextLine().toLowerCase(); // Convert the word to lowercase for simplicity**

**String vowels = "";**

**String consonants = "";**

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

**char ch = word.charAt(i);**

**if (isVowel(ch)) {**

**vowels += ch;**

**} else if (Character.isLetter(ch)) {**

**consonants += ch;**

**}**

**}**

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

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

**}**

**public static boolean isVowel(char ch) {**

**return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u';**

**}**

**}**

1. Write a program to print the Fibonacci series.

Sample Input:

Enter the n value: 6

Sample Output:

0 1 1 2 3 5

Test Condition: Implement negative Fibonacci series

**Program:**

**import java.util.Scanner;**

**public class FibonacciSeries {**

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

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

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

**int n = scanner.nextInt();**

**int[] fibonacciSeries = new int[n];**

**fibonacciSeries[0] = 0;**

**fibonacciSeries[1] = 1;**

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

**fibonacciSeries[i] = fibonacciSeries[i - 1] + fibonacciSeries[i - 2];**

**}**

**System.out.println("The Fibonacci series is: ");**

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

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

**}**

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

**}**

**}**

**28**.Write a program to print the below pattern

1

2 2

3 3 3

1. 4 4 4

**Program:**

**public class PatternPrinting {**

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

**int n = 4;**

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

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

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

**}**

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

**}**

**}**

**}**

1. 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

**Program:**

**import java.util.Scanner;**

**public class SquareAndCube {**

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

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

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

**if (scanner.hasNextDouble()) {**

**double number = scanner.nextDouble();**

**double square = number \* number;**

**double cube = number \* number \* number;**

**System.out.println("Square of " + number + " = " + square);**

**System.out.println("Cube of " + number + " = " + cube);**

**} else {**

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

**}**

**scanner.close();**

**}**

**}**

1. Program to find the frequency of each element in the array.

Sample Input & Output:

{1, 2, 8, 3, 2, 2, 2, 5, 1}

Element | Frequency

--------------------------

1 | 2

2 | 4

8 | 1

3 | 1

1. | 1

**Program:**

**import java.util.HashMap;**

**import java.util.Map;**

**public class ElementFrequency {**

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

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

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

**for (int element : array) {**

**if (frequencyMap.containsKey(element)) {**

**frequencyMap.put(element, frequencyMap.get(element) + 1);**

**} else {**

**frequencyMap.put(element, 1);**

**}**

**}**

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

**System.out.println("Element " + entry.getKey() + " occurs " + entry.getValue() + " times.");**

**}**

**}**

**}**

1. 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

**Program:**

**import java.util.Scanner;**

**public class PerfectNumber {**

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

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

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

**if (scanner.hasNextInt()) {**

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

**if (number > 0) {**

**int sum = 1;**

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

**if (number % i == 0) {**

**sum += i;**

**if (i != number / i) {**

**sum += number / i;**

**}**

**}**

**}**

**if (sum==number) {**

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

**} else {**

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

**}**

**} else {**

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

**}**

**} else {**

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

**}**

**scanner.close();**

**}**

**}**

1. Find the factorial of n?

Sample Input:

N = 6

Sample Output:

6 Factorial = 720

Test cases:

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

**32.**Write a program to print the below pattern

1

4 9

16 25 36

49 64 81 100

**Program:**

**public class PatternPrinting {**

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

**int n = 5;**

**int count = 1;**

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

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

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

**count++;**

**}**

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

**}**

**}**

**}**

1. 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}

**Program:**

**import java.util.Scanner;**

**public class CompositeNumberCounter {**

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

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

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

**int n = scanner.nextInt();**

**int[] array = new int[n];**

**System.out.println("Enter the elements:");**

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

**array[i] = scanner.nextInt();**

**}**

**int compositeCount = 0;**

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

**boolean isComposite = false;**

**for (int j = 2; j <= array[i] / 2; j++) {**

**if (array[i] % j == 0) {**

**isComposite = true;**

**break;**

**}**

**}**

**if (isComposite) {**

**compositeCount++;**

**}**

**}**

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

**}**

**}**

1. Find the nth odd number after n odd number

Sample Input:

N : 4

Sample Output:

4th Odd number after 4 odd numbers = 15

Test cases:

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

**Program:**

**import java.util.Scanner;**

**public class NthOddNumberAfterN {**

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

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

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

**if (scanner.hasNextInt()) {**

**int N = scanner.nextInt();**

**if (N >= 0) {**

**int result = calculateNthOddNumberAfterN(N);**

**System.out.println("The " + N + "th Odd number after " + N + " odd numbers = " + result);**

**} else {**

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

**}**

**} else {**

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

**}**

**scanner.close();**

**}**

**public static int calculateNthOddNumberAfterN(int N) {**

**return (2 \* N) + 7;**

**}**

**}**

1. 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.

**Program:**

**import java.util.Scanner;**

**public class CharacterSearchInString {**

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

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

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

**String inputString = scanner.nextLine();**

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

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

**scanner.close();**

**int index = findCharacter(inputString, searchChar);**

**if (index != -1) {**

**System.out.println("Character '" + searchChar + "' is present at index " + index);**

**} else {**

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

**}**

**}**

**public static int findCharacter(String str, char ch) {**

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

**if (str.charAt(i) == ch) {**

**return i;**

**}**

**}**

**return -1;**

**}**

**}**

1. Write a program to print the below pattern

1

2 2

3 3 3

4 4 4 4

3 3 3

2 2

1

**Program:**

**public class PatternPrinting {**

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

**int n = 5;**

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

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

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

**}**

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

**}**

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

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

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

**}**

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

**}**

**}**

**}**

1. 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

**Program:**

**import java.util.\*;**

**public class Armstrong {**

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

**Scanner sc=new Scanner(System.in);**

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

**int number=sc.nextInt();**

**int originalNumber, remainder, result = 0;**

**originalNumber = number;**

**while (originalNumber != 0)**

**{**

**remainder = originalNumber % 10;**

**result += remainder\*remainder\*remainder;**

**originalNumber /= 10;**

**}**

**if(result == number)**

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

**else**

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

**}**

**}**

1. 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
4. SATISFACTION
5. DEDICATION
6. 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

**Program:**

**import java.util.Scanner;**

**public class StringOperator{**

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

**{**

**String str1, str2;**

**Scanner scan = new Scanner(System.in);**

**System.out.print("Enter a String : ");**

**str1 = scan.nextLine();**

**str2 = str1.replaceAll("[aeiouAEIOU]", "");**

**System.out.print("All Vowels Removed Successfully..!!\nNew String is : ");**

**System.out.print(str2);**

**}**

**}**

1. Write a program to print hollow SquareDollar pattern?

**Program:**

**import java.util.Scanner;**

**public class HollowSquare1 {**

**private static Scanner sc;**

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

**sc = new Scanner(System.in);**

**System.out.print("Enter Hollow Square Side = ");**

**int side = sc.nextInt();**

**System.out.println("Printing Hollow Square Star Pattern");**

**for (int i = 0; i < side; i++ )**

**{**

**for (int j = 0 ; j < side; j++ )**

**{**

**if (i == 0 || i == side - 1 || j == 0 || j == side - 1)**

**{**

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

**}**

**else {**

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

**}**

**}**

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

**}**

**}**

**}**

1. 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 number: 143

Sample Output:

Sum of 3 digit number: 8

Test cases:

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

**Program:**

**import java.util.Scanner;**

**public class SumOfDigits {**

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

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

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

**int n = scanner.nextInt();**

**if (n >= 1) {**

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

**int num = scanner.nextInt();**

**int sum = 0;**

**while (num > 0) {**

**sum += num % 10;**

**num /= 10;**

**}**

**while (sum >= 10) {**

**int tempSum = 0;**

**while (sum > 0) {**

**tempSum += sum % 10;**

**sum /= 10;**

**}**

**sum = tempSum;**

**}**

**System.out.println("Sum of " + n + " digit number: " + sum);**

**}**

**}**

**}**

1. 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

**Program:**

**import java.util.Scanner;**

**public class SquareRoot {**

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

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

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

**if (scanner.hasNextInt()) {**

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

**if (number >= 0) {**

**double sqrt = Math.sqrt(number);**

**System.out.println("Positive square root: " + sqrt);**

**System.out.println("Negative square root: " + (-sqrt));**

**} else {**

**System.out.println("Please enter a non-negative perfect square number.");**

**}**

**} else {**

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

**}**

**}}**

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

22 18

1. Write a program to print inverted pyramid pattern.

**Program:**

**public class Main {**

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

**int rows = 5;**

**for(int i = rows; i >= 1; --i) {**

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

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

**}**

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

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

**}**

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

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

**}**

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

**}**

**}**

**}**