Java Easy Answers

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

**Sample Input:**

**String: TEMPLE**

**Sample Output:**

**Reverse String: ELPMET**

import java.util.\*;

public class reversestring{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

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

String str=sc.nextLine();

String reverse = "";

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

reverse=reverse+str.charAt(i);

}

System.out.println("Reverse of"+str+"is:"+reverse);

}

}

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

import java.util.\*;

class uservalid{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

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

String a=sc.nextLine();

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

String b=sc.nextLine();

if(a.equals(b)){

System.out.println("user is valid");

}else{

System.out.println("user is invalid");

}

}

}

**3.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**

import java.util.\*;

class ReverseNumber{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

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

int number=sc.nextInt();

int reverse = 0;

while(number != 0)

{

int remainder = number % 10;

reverse = reverse \* 10 + remainder;

number = number/10;

}

System.out.println("The reverse of the given number is: " + reverse);

} }

**4.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**

import java.util.\*;

class voteeligibility{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

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

int age=sc.nextInt();

if(age>=18){

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

}

else if(age<0){

System.out.println("Invalid input");

}

else{

System.out.println("you are allowed to vote after"+(18-age)+"years");

}

}

}

**5.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}**

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);

}

}

**6.Write a program to print Right Triangle Star Pattern**

**Sample Input: n = 5**

**Output:**

**\***

**\* \***

**\* \* \***

**\* \* \* \***

class pyramid{

public static void main(String[] args){

int row,i,j;

row=5;

System.out.println("");

for(i=1;i<=row; i++){

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

System.out.println(" ");

}

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

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

}

System.out.println(" ");

}

}

}

**7.Write a program to print the below pattern?**

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

public class Main {

public static void main(String[] args) {

int rows = 6, coef = 1;

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

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

System.out.print(" ");

}

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

if (j == 0 || i == 0)

coef = 1;

else

coef = coef \* (i - j + 1) / j;

System.out.printf("%4d", coef);

}

System.out.println();

}

}

}

**8.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**

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);

}

}

**9.[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)**

import java.io.\*;

class sumevenfib {

static int Fib\_Even\_Sum(int N)

{

if (N <= 0)

return 0;

int fib[] = new int[2 \* N + 1];

fib[0] = 0;

fib[1] = 1;

int s = 0;

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

fib[j] = fib[j - 1] + fib[j - 2];

if (j % 2 == 0)

s += fib[j];

}

return s;

}

public static void main(String[] args)

{

int N = 4;

System.out.println(

"Even sum of fibonacci series till number " + N

+ " is: " + +Fib\_Even\_Sum(N));

}

}

**10. 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**

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();

}

}

**11.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**

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();

}

}}

**12.Write a program to print rectangle symbol pattern.**

**Get the symbol as input from user**

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();

}

}

**13.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**

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();

}

}

**14.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**

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();

}

}}

**15.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**

public class PatternPrint {

public static void main(String[] args) {

int n = 3;

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

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

System.out.print("1 ");

}

System.out.println();

}

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

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

System.out.print("1 ");

}

System.out.println();

}

}

}

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

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();

}

}

**17.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**

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;

}

}

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

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();

}

}

}

**19.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}**

import java.util.\*;

public class MeanMedianMode {

public static void main(String[] args) {

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

double mean = Arrays.stream(arr).average().orElse(Double.NaN);

double median = calculateMedian(arr);

int mode = calculateMode(arr);

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

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

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

}

public static double calculateMedian(int[] arr) {

Arrays.sort(arr);

int middle = arr.length / 2;

return (arr.length % 2 == 0) ? (arr[middle - 1] + arr[middle]) / 2.0 : arr[middle];

}

public static int calculateMode(int[] arr) {

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

int maxFrequency = 0, mode = 0;

for (int num : arr) {

int currentFrequency = frequencyMap.merge(num, 1, Integer::sum);

if (currentFrequency > maxFrequency) {

maxFrequency = currentFrequency;

mode = num;

}

}

return mode;

}

}

**20.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**

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();

}

}

**21.Write a program to print the following pattern**

**Sample Input:**

**Enter the Character to be printed: %**

**Max Number of time printed: 3**

**%**

**% %**

**% % %**

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();

}

}

}

**22.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**

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

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 dateString = scanner.nextLine();

scanner.close();

SimpleDateFormat dateFormat = new SimpleDateFormat("MM/dd/yyyy");

try {

Date date = dateFormat.parse(dateString);

int year = getYear(date);

if (isLeapYear(year)) {

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

} else {

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

}

} catch (ParseException e) {

System.out.println("Invalid date format. Please enter the date in MM/dd/yyyy format.");

}

}

public static int getYear(Date date) {

SimpleDateFormat yearFormat = new SimpleDateFormat("yyyy");

return Integer.parseInt(yearFormat.format(date));

}

public static boolean isLeapYear(int year) {

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

return true;

} else {

return false;

}

}

}

**23.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**

import java.util.Scanner;

public class FactorsCount {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a positive integer: ");

int number = scanner.nextInt();

scanner.close();

if (number <= 0) {

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

} else {

int count = countFactors(number);

System.out.println("The number of factors for " + number + " is " + count);

}

}

public static int countFactors(int n) {

int count = 0;

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

if (n % i == 0) {

count++;

}

}

return count;

}

}

**24.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**

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();

}

}

**25.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**

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);

}

}

**26.Write a program to print hollow square symbol pattern?**

**Get the symbol from user.**

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();

}

}

}

**27.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**

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();

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';

}

}

**28.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**

public class PositiveNegativeFibonacci {

public static void main(String[] args) {

int n = 10;

System.out.println("Positive Fibonacci Series:");

printFibonacci(n, false);

System.out.println("\nNegative Fibonacci Series:");

printFibonacci(n, true);

}

public static void printFibonacci(int n, boolean negative) {

int a = 0, b = 1, c;

int sign = negative ? -1 : 1;

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

System.out.print(sign \* a + " ");

c = a + b;

a = b;

b = c;

}

}

}

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

**1**

**2 2**

**3 3 3**

1. **4 4 4**

public class NumberPattern {

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();

}

}

}

**30.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**

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();

}

}

**31.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**

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.");

}

}

}

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

**1**

**4 9**

**16 25 36**

**49 64 81 100**

public class NumberPattern {

public static void main(String[] args) {

int rows = 4;

int count = 1;

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

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

System.out.print(" ");

}

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

System.out.printf("%4d", count \* count);

count++;

}

System.out.println();

}

}

}

**33.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}**

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);

}

}

**34.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**

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;

}

}

**35.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.**

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;

}

}

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

**1**

**2 2**

**3 3 3**

**4 4 4 4**

**3 3 3**

**2 2**

**1**

public class NumberPattern {

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();

}

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

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

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

}

System.out.println();

}

}

}

**37.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**

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.");

}

}

**38.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**

import java.util.Arrays;

import java.util.Scanner;

public class ReverseAlphabeticalOrder {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String word = scanner.nextLine();

char[] characters = word.toCharArray();

Arrays.sort(characters);

for (int i = 0; i < characters.length / 2; i++) {

char temp = characters[i];

characters[i] = characters[characters.length - 1 - i];

characters[characters.length - 1 - i] = temp;

}

String result = new String(characters);

System.out.println("Word in Reverse Alphabetical Order: " + result);

}

}

**39.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**

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);

}

}

**40.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**

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);

}

}

}

**41.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**

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.");

}

scanner.close();

}

}

**42.Write a program to print inverted pyramid pattern.**

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();

}

}

}