**JAVA MEDIUM ANSWERS**

1. **Write a program to count all the prime and composite numbers entered by the user.**

**Sample Input:**

**Enter the numbers**

**4**

**54**

**29**

**71**

**7**

**59**

**98**

**23**

**Sample Output:**

**Composite number:3**

**Prime number:5**

**Test cases:**

1. **33, 41, 52, 61,73,90**
2. **TEN, FIFTY, SIXTY-ONE, SEVENTY-SEVEN, NINE**
3. **45, 87, 09, 5.0 ,2.3, 0.4**
4. **-54, -76, -97, -23, -33, -98**
5. **45, 73, 00, 50, 67, 44**

import java.util.Scanner;

public class PrimeCompositeCounter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int primeCount = 0, compositeCount = 0;

System.out.println("Enter numbers (non-integer to finish):");

while (scanner.hasNextInt()) {

int n = scanner.nextInt();

if (n < 2 || !isPrime(n)) compositeCount++;

else primeCount++;

}

System.out.printf("Composite numbers: %d%nPrime numbers: %d%n", compositeCount, primeCount);

}

public static boolean isPrime(int n) {

if (n <= 3) return n > 1;

if (n % 2 == 0 || n % 3 == 0) return false;

for (int i = 5; i \* i <= n; i += 6) {

if (n % i == 0 || n % (i + 2) == 0) return false;

}

return true;

}

}

1. **Find the Mth maximum number and Nth minimum number in an array and then find the sum of it and difference of it.**

**Sample Input:**

**Array of elements = {14, 16, 87, 36, 25, 89, 34}**

**M = 1**

**N = 3**

**Sample Output:**

**1stMaximum Number = 89**

**3rdMinimum Number = 25**

**Sum = 114**

**Difference = 64**

**Test cases:**

1. **{16, 16, 16 16, 16}, M = 0, N = 1**
2. **{0, 0, 0, 0}, M = 1, N = 2**
3. **{-12, -78, -35, -42, -85}, M = 3 , N = 3**
4. **{15, 19, 34, 56, 12}, M = 6 , N = 3**
5. **{85, 45, 65, 75, 95}, M = 5 , N = 7**

import java.util.Scanner;

public class maxmin{

public static void main(String[] args){

Scanner input=new Scanner(System.in);

System.out.print("Enter the array size: ");

int s=input.nextInt();

int i,j,temp;

int a[]=new int[s];

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

for(i=0;i<s;i++){

a[i]=input.nextInt();

}

Scanner sc=new Scanner(System.in);

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

int m=sc.nextInt();

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

int n=sc.nextInt();

for(i=0;i<s;i++){

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

if(a[i]>a[j]){

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

System.out.println(m+" maximum element is: "+(a[s-m]));

System.out.println(n+" minimum element is: "+a[n-1]);

System.out.println("Sum = "+(a[s-m]+a[n-1]));

System.out.println("Difference" = "+(a[s-m]-a[n-1]));

}

}

1. **Write a program to print the total amount available in the ATM machine with the conditions applied.**

**Total denominations are 2000, 500, 200, 100, get the denomination priority from the user and the total number of notes from the user to display the total available balance to the user**

**Sample Input:**

**Enter the 1st Denomination: 500**

**Enter the 1st Denomination number of notes: 4**

**Enter the 2nd Denomination: 100**

**Enter the 2nd Denomination number of notes: 20**

**Enter the 3rd Denomination: 200**

**Enter the 3rd Denomination number of notes: 32**

**Enter the 4th Denomination: 2000**

**Enter the 4th Denomination number of notes: 1**

**Sample Output:**

**Total Available Balance in ATM: 12400**

**Test Cases:**

**3 Hidden Test cases (Think Accordingly based on Denominations)**

import java.util.Scanner;

public class ATMTotalBalance {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int[] denominations = new int[4];

int[] notes = new int[4];

for (int i = 0; i < 4; i++) {

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

denominations[i] = scanner.nextInt();

System.out.print("Enter the " + (i + 1) + " Denomination number of notes: ");

notes[i] = scanner.nextInt();

}

int totalBalance = calculateTotalBalance(denominations, notes);

System.out.println("Total Available Balance in ATM: " + totalBalance);

}

public static int calculateTotalBalance(int[] denominations, int[] notes) {

int totalBalance = 0;

for (int i = 0; i < 4; i++) {

totalBalance += denominations[i] \* notes[i];

}

return totalBalance;

}

}

1. **Write a program using choice to check**

**Case 1: Given string is palindrome or not**

**Case 2: Given number is palindrome or not**

**Sample Input:**

**Case = 1**

**String = MADAM**

**Sample Output:**

**Palindrome**

**Test cases:**

1. **MONEY**
2. **5678765**
3. **MALAY12321ALAM**
4. **MALAYALAM**
5. **1234.4321**

import java.util.Scanner;

public class PalindromeChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Palindrome Checker Menu:");

System.out.println("1. Check if a string is a palindrome");

System.out.println("2. Check if a number is a palindrome");

System.out.print("Enter your choice (1 or 2): ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

checkPalindrome("string", scanner);

break;

case 2:

checkPalindrome("number", scanner);

break;

default:

System.out.println("Invalid choice. Please enter 1 or 2.");

}

}

public static void checkPalindrome(String type, Scanner scanner) {

System.out.print("Enter a " + type + ": ");

String input = scanner.next();

boolean isPalindrome = true;

for (int i = 0, j = input.length() - 1; i < j; i++, j--) {

if (input.charAt(i) != input.charAt(j)) {

isPalindrome = false;

break;

}

}

System.out.println("The " + type + " is " + (isPalindrome ? "a palindrome." : "not a palindrome."));

}

}

1. **Write a program to convert Decimal number equivalent to Binary number and octal numbers?**

**Sample Input:**

**Decimal Number: 15**

**Sample Output:**

**Binary Number = 1111**

**Octal = 17**

**Test cases:**

1. **111**
2. **15.2**
3. **0**
4. **B12**
5. **1A.2**

import java.util.Scanner;

public class DecimalToBinaryOctal {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int decimalNumber = scanner.nextInt();

String binaryNumber = Integer.toBinaryString(decimalNumber);

String octalNumber = Integer.toOctalString(decimalNumber);

System.out.println("Binary Number = " + binaryNumber);

System.out.println("Octal = " + octalNumber);

}}

1. **In an organization they decide to give bonus to all the employees on New Year. A 5% bonus on salary is given to the grade A workers and 10% bonus on salary to the grade B workers. Write a program to enter the salary and grade of the employee. If the salary of the employee is less than $10,000 then the employee gets an extra 2% bonus on salary Calculate the bonus that has to be given to the employee and print the salary that the employee will get.**

**Sample Input & Output:**

**Enter the grade of the employee: B**

**Enter the employee salary: 50000**

**Salary=50000**

**Bonus=5000.0**

**Total to be paid:55000.0**

**Test cases:**

1. **Enter the grade of the employee: A**

**Enter the employee salary: 8000**

1. **Enter the grade of the employee: C**

**Enter the employee salary: 60000**

1. **Enter the grade of the employee: B**

**Enter the employee salary: 0**

1. **Enter the grade of the employee: 38000**

**Enter the employee salary: A**

1. **Enter the grade of the employee: B**

**Enter the employee salary: -8000**

import java.util.Scanner;

public class EmployeeBonus {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the grade of the employee: ");

String grade = scanner.nextLine();

System.out.println("Enter the employee salary: ");

double salary = scanner.nextDouble();

double bonus = 0;

if (grade.equals("A")) {

bonus = salary \* 0.05;

} else if (grade.equals("B")) {

bonus = salary \* 0.1;

}

if (salary < 10000) {

bonus += salary \* 0.02;

}

System.out.println("Bonus=" + bonus);

System.out.println("Total to be paid:" + (salary + bonus));

}

}

1. **Write a program to print the first n perfect numbers. (Hint Perfect number means a positive integer that is equal to the sum of its proper divisors)**

**Sample Input:**

**N = 3**

**Sample Output:**

**First 3 perfect numbers are: 6 , 28 , 496**

**Test Cases:**

1. **N = 0**
2. **N = 5**
3. **N = -2**
4. **N = -5**

**5.N = 0.2**

import java.util.Scanner;

public class PerfectNumbers {

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("Enter a positive integer.");

return;

}

System.out.print("First " + n + " perfect numbers are: ");

int count = 0, number = 2;

while (count < n) {

if (isPerfectNumber(number)) {

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

count++;

}

number++;

}

}

public static boolean isPerfectNumber(int num) {

int sum = 1;

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

if (num % i == 0) {

sum += i;

if (i != num / i) {

sum += num / i;

}

}

}

return sum == num;

}

}

1. **Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.**

**Sample Input & Output:**

**Enter the marks in python: 90**

**Enter the marks in c programming: 91**

**Enter the marks in Mathematics: 92**

**Enter the marks in Physics: 93**

**Total= 366**

**Aggregate = 91.5**

**DISTINCTION**

**Test cases:**

1. **18, 76,93,65**
2. **73,78,79,75**
3. **98,106,120,95**
4. **96,73, -85,95**
5. **78,59.8,76,79**

import java.util.\*;

public class StudentMarks {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the marks in python: ");

int python = scanner.nextInt();

System.out.println("Enter the marks in c programming: ");

int cProgramming = scanner.nextInt();

System.out.println("Enter the marks in Mathematics: ");

int mathematics = scanner.nextInt();

System.out.println("Enter the marks in Physics: ");

int physics = scanner.nextInt();

int total = python + cProgramming + mathematics + physics;

double aggregate = total / 4.0;

System.out.println("Total= " + total);

System.out.println("Aggregate = " + aggregate);

if (aggregate >= 75) {

System.out.println("DISTINCTION");

} else if (aggregate >= 60) {

System.out.println("FIRST DIVISION");

} else if (aggregate >= 50) {

System.out.println("SECOND DIVISION");

} else if (aggregate >= 40) {

System.out.println("THIRD DIVISION");

} else {

System.out.println("FAIL");

}

}

}

1. **Write a program to calculate tax given the following conditions:**
   1. **If income is less than or equal to 1,50,000 then no tax**
   2. **If taxable income is 1,50,001 – 3,00,000 the charge 10% tax**
   3. **If taxable income is 3,00,001 – 5,00,000 the charge 20% tax**
   4. **If taxable income is above 5,00,001 then charge 30% tax**

**Sample Input:**

**Enter the income:200000**

**Sample Output:**

**Tax= 20000**

**Test cases:**

1. **400700**
2. **2789239**
3. **150000**
4. **00000**
5. **-125486**

import java.util.Scanner;

public class TaxCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the income: ");

int income = scanner.nextInt();

double tax = 0;

if (income <= 150000) {

tax = 0;

} else if (income>150000 && income < 300000) {

tax = income\* 0.1;

} else if (income>300000 && income < 500000) {

tax = income \* 0.2;

} else {

tax = income\* 0.3;

}

System.out.println("Tax= " + tax);

}

}

1. **Write a 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:**

1. **M = 6, N = -3**
2. **M = -3, N = 5**
3. **M = 4, N = 0**
4. **M = 0, N = 0**
5. **M = -5, N = -5**

import java.util.\*;

public class MultiplicationTable {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int m = scanner.nextInt();

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

int n = scanner.nextInt();

System.out.println("Multiplication Table for " + m + " up to " + n + ":");

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

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

}

}

}

1. **Write a program to read the numbers until -1 is encountered. Find the average of positive numbers and negative numbers entered by user.**

**Sample Input:**

**Enter -1 to exit…**

**Enter the number: 7**

**Enter the number: -2**

**Enter the number: 9**

**Enter the number: -8**

**Enter the number: -6**

**Enter the number: -4**

**Enter the number: 10**

**Enter the number: -1**

**Sample Output:**

**The average of negative numbers is: -5.0**

**The average of positive numbers is : 8.66666667**

**Test cases:**

1. **-1,43, -87, -29, 1, -9**
2. **73, 7-6,2,10,28,-1**
3. **-5, -9, -46,2,5,0**
4. **9, 11, -5, 6, 0,-1**
5. **-1,-1,-1,-1,-1**

import java.util.Scanner;

public class AverageCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int posCount = 0, negCount = 0;

double posSum = 0.0, negSum = 0.0;

System.out.println("Enter numbers (enter -1 to stop):");

int num;

while ((num = scanner.nextInt()) != -1) {

if (num > 0) {

posCount++;

posSum += num;

} else if (num < 0) {

negCount++;

negSum += num;

}

}

if (posCount > 0)

System.out.println("The average of positive numbers is: " + (posSum / posCount));

else

System.out.println("No positive numbers entered.");

if (negCount > 0)

System.out.println("The average of negative numbers is: " + (negSum / negCount));

else

System.out.println("No negative numbers entered.");

}

}

1. **Write a program to read a character until a \* is encountered. Also count the number of uppercase, lowercase, and numbers entered by the users.**

**Sample Input:**

**Enter \* to exit…**

**Enter any character: W**

**Enter any character: d**

**Enter any character: A**

**Enter any character: G**

**Enter any character: g**

**Enter any character: H**

**Enter any character: \***

**Sample Output:**

**Total count of lower case:2**

**Total count of upper case:4**

**Total count of numbers =0**

**Test cases:**

1. **1,7,6,9,5**
2. **S, Q, l, K,7, j, M**
3. **M, j, L, &, @, G**
4. **D, K, I, 6, L, \***
5. **\*, K, A, e, 1, 8, %, \***

import java.util.Scanner;

public class CharacterCounter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int uppercaseCount = 0;

int lowercaseCount = 0;

int digitCount = 0;

System.out.println("Enter characters (stop with '\*'):");

while (true) {

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

if (input == '\*') {

break;

}

if (Character.isUpperCase(input)) {

uppercaseCount++;

} else if (Character.isLowerCase(input)) {

lowercaseCount++;

} else if (Character.isDigit(input)) {

digitCount++;

}

}

System.out.println("Uppercase letters: " + uppercaseCount);

System.out.println("Lowercase letters: " + lowercaseCount);

System.out.println("Digits: " + digitCount);

}

}

1. **Write a program to calculate the factorial of number using recursive function.**

**Sample Input & Output:**

**Enter the value of n: 6**

**Sample Input & Output:**

**The factorial of 6 is: 720**

**Test cases:**

1. **N = 0**
2. **N = -5**
3. **N = 1**
4. **N = M**
5. **N = %**

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 Find the Nth Largest Number in a array.**

**Sample Input:**

**List : {14, 67, 48, 23, 5, 62}**

**N = 4**

**Sample Output:**

**4th Largest number: 23**

**Test cases:**

1. **N = 0**
2. **N = -5**
3. **N = 1**
4. **N = M**
5. **N = %**

import java.util.Scanner;

public class maxmin{

public static void main(String[] args){

Scanner input=new Scanner(System.in);

System.out.print("Enter the array size: ");

int s=input.nextInt();

int i,j,temp;

int a[]=new int[s];

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

for(i=0;i<s;i++){

a[i]=input.nextInt();

}

Scanner sc=new Scanner(System.in);

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

int m=sc.nextInt();

for(i=0;i<s;i++){

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

if(a[i]>a[j]){

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

System.out.println(m+" maximum element is: "+(a[s-m]));

}

}

1. **Write a program to find the number of special characters in the given statement**

**Sample Input:**

**Given statement: Modi Birthday @ September 17, #&$% is the wishes code for him.**

**Sample Output:**

**Number of special Characters: 5**

public class SpecialCharacterCounter {

public static void main(String[] args) {

String statement = "Modi Birthday @ September 17, #&$% is the wishes code for him.";

int specialCharacterCount = countSpecialCharacters(statement);

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

}

public static int countSpecialCharacters(String input) {

int count = 0;

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

char c = input.charAt(i);

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

count++;

}

}

return count;

}

}

1. **Write a Program to Remove the Duplicate Items from a array.**

**Sample Input:**

**Enter the number of elements in array:7**

**Enter element1:10**

**Enter element2:20**

**Enter element3:20**

**Enter element4:30**

**Enter element5:40**

**Enter element6:40**

**Enter element7:50**

**Sample Output:**

**Non-duplicate items:**

**[10, 20, 30, 40, 50]**

import java.util.Arrays;

public class RemoveDuplicatesFromArray {

public static void main(String[] args) {

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

int[] uniqueArr = Arrays.stream(arr).distinct().toArray();

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

System.out.println("Array with Duplicates Removed: " + Arrays.toString(uniqueArr));

}

}

1. **Write a program to convert the Binary to Decimal, Octal**

**Sample Input:**

**Given Number: 1101**

**Sample Output:**

**Decimal Number: 13**

**Octal:15**

**Test cases:**

1. **211**
2. **11011**
3. **22122**
4. **111011.011**
5. **1010.0101**

import java.util.Scanner;

public class BinaryConversion {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

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

String b = s.next();

s.close();

if (b.matches("[01]+")) {

int d = 0, l = b.length();

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

if (b.charAt(i) == '1')

d += Math.pow(2, l - 1 - i);

}

System.out.println("Decimal Number: " + d);

System.out.println("Octal: " + Integer.toOctalString(d));

} else {

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

}

}

}

1. **Bank is a class that provides method to get the rate of interest. But, rate of interest may differ according to banks. For example, SBI, ICICI and AXIS banks are providing 8.4%, 7.3% and 9.7% rate of interest. Write a Java program for above scenario.**

**Input :**

**Assign rate of interest for 3 banks**

**Pseudo**

**Define the Super class Bank , in which getRateOfInterest() as overriding method**

**Define subclass SBI, ICICI, AXIS , in which return the rate of interest as per given values respectively**

**Declare object and print the rate of interest for the three banks**

**Output:**

**Sample Input SBI, 8.4**

**Sample Output**

**Test case**

1. **SBI, 8.3**
2. **ICICI, 7.3**
3. **AXIS, 9.7**
4. **SBI, 8.6**
5. **AXIX, 7.6**

class Bank {

public double getRateOfInterest() {

return 0.0;

}

}

class SBI extends Bank {

@Override

public double getRateOfInterest() {

return 8.4;

}

}

class ICICI extends Bank {

@Override

public double getRateOfInterest() {

return 7.3;

}

}

class AXIS extends Bank {

@Override

public double getRateOfInterest() {

return 9.7;

}

}

class bank {

public static void main(String[] args) {

SBI sbi = new SBI();

ICICI icici = new ICICI();

AXIS axis = new AXIS();

System.out.println("1. SBI, " + sbi.getRateOfInterest());

System.out.println("2. ICICI, " + icici.getRateOfInterest());

System.out.println("3. AXIS, " + axis.getRateOfInterest());

SBI sbi2 = new SBI();

System.out.println("4. SBI, " + sbi2.getRateOfInterest());

}

}

1. **Bring out the situation in which member names of a subclass hide members by the same name in the super class. How it can be resolved? Write Suitable code in Java and**

**Implement above scenario with the Parametrized Constructor (accept int type parameter) of the Super Class can be called from Sub Class Using super () and display the input values provided.**

**Input :**

**Assign or input values for super class and sub class members.**

**Pseudo :**

**Define super class and sub class with one member (has same name)**

**Define method in super class and sub class with same method signature**

**Declare the object in main method**

**Invoke methods using object to display the values**

**Output :**

**Sample Input : 100, 200**

**Sample Output : 100, 200**

**Test Cases**

1. **10, 20**
2. **-20, -30**
3. **0, 0**
4. **EIGHT FIVE**

**5.10.57, 12.58**

class SuperClass {

int num;

public SuperClass(int num) {

this.num = num;

}

public void display() {

System.out.println("SuperClass: " + num);

}

}

class SubClass extends SuperClass {

int num;

public SubClass(int num, int subNum) {

super(num);

this.num = subNum;

}

@Override

public void display() {

super.display();

System.out.println("SubClass: " + num);

}

}

class MemberHidingExample {

public static void main(String[] args) {

SubClass sub1 = new SubClass(10, 20);

sub1.display();

SubClass sub2 = new SubClass(-20, -30);

sub2.display();

SubClass sub3 = new SubClass(0, 0);

sub3.display();

SubClass sub4 = new SubClass(57, 12);

sub4.display();

}

}

1. **Display Multiplication table for 5 and 10 using various stages of life cycle of the thread by generating a suitable code in Java.**

**Sample Input 5, 10**

**5 X 1 = 5**

**5 X 2 =10**

**….**

**10 X 1 =10**

**10 X 2 = 20**

**….**

**Test Cases:**

1. **10, 20**
2. **-10, -30**
3. **0, 0**
4. **SIX, SIX**
5. **9.8, 9.6**

import java.util.\*;

class tables{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

int m=5,k=10;

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

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

}

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

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

}

}

}

1. **Using the concepts of thread with implementing Runnable interface in Java to generate Fibonacci series.**

**Sample Input : 5**

**Sample Output : 0 1 1 2 3 …..**

**Test Cases**

1. **7**
2. **-10**
3. **0**
4. **EIGHT FIVE**
5. **12.65**

import java.util.Scanner;

public class FibonacciSeries {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of terms in the Fibonacci series: ");

int n = scanner.nextInt();

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

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

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

}

scanner.close();

}

private static int fibonacci(int n) {

if (n <= 1) {

return n;

} else {

return fibonacci(n - 1) + fibonacci(n - 2);

}

}

}

1. **Generate a Java code to find the sum of N numbers using array and throw ArrayIndexOutOfBoundsException when the loop variable beyond the size N.**

**Sample Input : 5**

**1 2 3 4 5**

**Sample Output : 15**

**Test Cases**

1. **4, 10**
2. **-10**
3. **0**
4. **EIGHT SEVEN**
5. **12.68**

import java.util.\*;

class R192110328{

public static void main(String[] args){

int n=10,a=0,b=0;

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

if(n>0)

a=n\*(n+1);

b=a/2;

}

System.out.println("sum of "+n+" numbers = "+b);

}

1. }**Using the concepts of thread with implementing Runnable interface in Java to find whether a given number is prime or not.**

**Sample Input : 5**

**Sample Output : 5 is Prime**

**Sample Output : 15**

**Test Cases**

1. **4**
2. **-10**
3. **0**
4. **EIGHT SEVEN**
5. **11.48**

import java.util.Scanner;

public class PrimeNumberChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (number<= 1) {

System.out.print("please enter a positive number");

}

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

if (number % i == 0) {

System.out.println(number + " is not a prime number.");

}

else{

System.out.println(number + " is a prime number.");

}

}

}

}

**24.Given a string s consisting of words and spaces, return the length of the last word in the string. A word is a maximal substring consisting of non-space characters only. There will be at least one word, consists of only English letters and spaces ' '.**

**Example 1:**

**Input:** s = "Hello World"

**Output:** 5

**Explanation:** The last word is "World" with length 5.

**Test Case**

|  |  |
| --- | --- |
| **Test Case** | **Inputs-1** |
|  | Maximal Substring Consisting |
|  | **lea@st one wor2d** |
|  | 1254 98076 |
|  | & \* ( ) % # $ |
|  | letters and spaces |

class LastWordLength {

public static void main(String[] args) {

System.out.println(lengthOfLastWord("Maximal Substring Consisting"));

}

public static int lengthOfLastWord(String s) {

s = s.trim();

int length = 0;

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

if (s.charAt(i) == ' ') {

break;

}

length++;

}

return length;

}

}