31)

import java.util.Scanner;

public class LCMandGCD {

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[] numbers = new int[n];

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

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

numbers[i] = scanner.nextInt();

}

int lcm = calculateLCM(numbers);

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

int gcd = calculateGCD(numbers);

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

scanner.close();

}

private static int calculateLCM(int[] numbers) {

int lcm = numbers[0];

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

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

}

return lcm;

}

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

if (b == 0) {

return a;

}

return calculateGCD(b, a % b);

}

private static int calculateGCD(int[] numbers) {

int gcd = numbers[0];

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

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

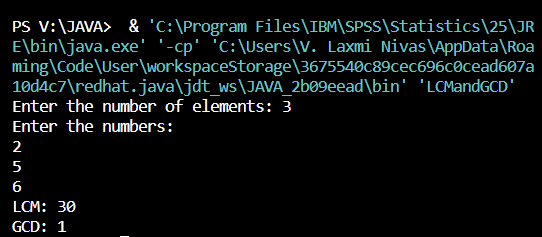
}

return gcd;

}

}

Output:



32)

import java.util.Scanner;

public class SimpleInterestCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

double principal = scanner.nextDouble();

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

int years = scanner.nextInt();

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

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

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

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

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

scanner.close();

}

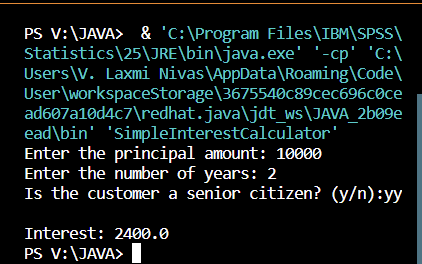
private static double calculateSimpleInterest(double principal, double interestRate, int years) {

return principal \* interestRate \* years;

}

}

Output:



33)

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:");

printFibonacciSeries(n);

scanner.close();

}

private static void printFibonacciSeries(int n) {

int firstTerm = 0, secondTerm = 1;

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

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

int nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

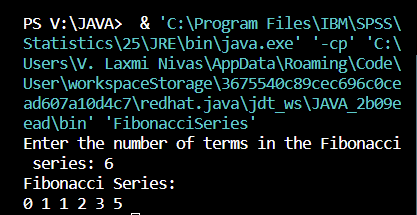
secondTerm = nextTerm;

}

}

}

Output:



34)

import java.util.Scanner;

public class EvenSumFibonacci {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

long evenSum = calculateEvenSumFibonacci(n);

System.out.println("Sum of even Fibonacci numbers up to " + n + ": " + evenSum);

scanner.close();

}

private static long calculateEvenSumFibonacci(int n) {

long firstTerm = 0, secondTerm = 1, evenSum = 0;

while (firstTerm <= n) {

if (firstTerm % 2 == 0) {

evenSum += firstTerm;

}

long nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

secondTerm = nextTerm;

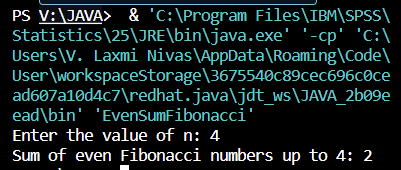
}

return evenSum;

}

}

Output:



35)

import java.util.Scanner;

public class SkipNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int m = scanner.nextInt();

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

int n = scanner.nextInt();

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

int k = scanner.nextInt();

System.out.println("Numbers from " + m + " to " + n + " by skipping " + k + " numbers:");

printNumbersWithSkip(m, n, k);

scanner.close();

}

private static void printNumbersWithSkip(int m, int n, int k) {

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

System.out.print(i);

if (i + k <= n) {

System.out.print(", ");

}

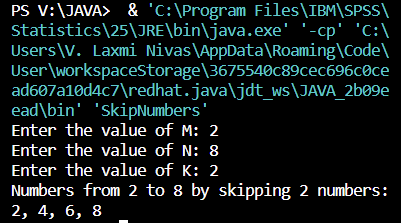
}

System.out.println();

}

}

Output:



36)

import java.util.Scanner;

public class CompositeNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int a = scanner.nextInt();

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

int b = scanner.nextInt();

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

printCompositeNumbers(a, b);

scanner.close();

}

private static boolean isComposite(int num) {

if (num <= 1) {

return false;

}

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

if (num % i == 0) {

return true;

}

}

return false;

}

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

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

if (isComposite(i)) {

System.out.print(i);

if (i < b) {

System.out.print(", ");

}

}

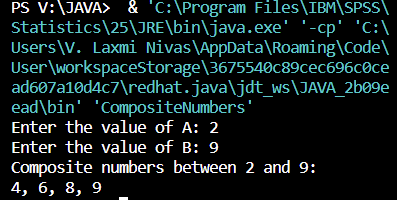
}

System.out.println();

}

}

Output:



37)

import java.util.Scanner;

public class FactorialCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

long factorial = calculateFactorial(n);

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

scanner.close();

}

private static long calculateFactorial(int n) {

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

return 1;

}

long result = 1;

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

result \*= i;

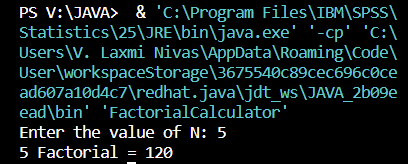
}

return result;

}

}

Output:



38)

import java.util.Scanner;

public class LeapYearChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String date = scanner.nextLine();

int year = extractYear(date);

boolean isLeapYear = checkLeapYear(year);

if (isLeapYear) {

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

} else {

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

}

scanner.close();

}

private static int extractYear(String date) {

String[] parts = date.split("/");

return Integer.parseInt(parts[2]);

}

private static boolean checkLeapYear(int year) {

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

return true;

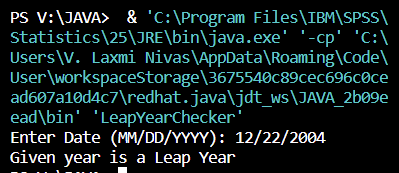
}

return false;

}

}

Output:



39)

import java.util.Scanner;

public class NumberOfFactors {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

int numOfFactors = countFactors(number);

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

scanner.close();

}

private static int countFactors(int number) {

int count = 0;

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

if (number % i == 0) {

if (number / i == i) {

count++;

} else {

count += 2;

}

}

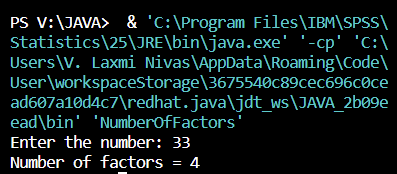
}

return count;

}

}

Output:



40)  
import java.util.Scanner;

public class PerfectNumberChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

boolean isPerfect = checkPerfectNumber(number);

if (isPerfect) {

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

} else {

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

}

scanner.close();

}

private static boolean checkPerfectNumber(int number) {

if (number <= 1) {

return false;

}

int sum = 1;

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

if (number % i == 0) {

sum += i;

if (i != number / i) {

sum += number / i;

}

}

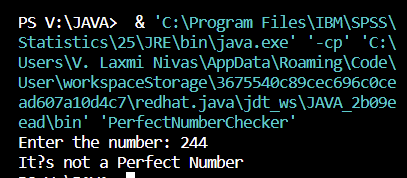
}

return sum == number;

}

}

Output:



41)

import java.util.Scanner;

public class SquareAndCubeCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

double decimalNumber = scanner.nextDouble();

double square = calculateSquare(decimalNumber);

double cube = calculateCube(decimalNumber);

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

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

scanner.close();

}

private static double calculateSquare(double number) {

return number \* number;

}

private static double calculateCube(double number) {

return number \* number \* number;

}

}

Output:

