1. **Check the given number is EVEN or ODD.**

import java.util.\*;

class Cdac{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

if(num%2==0){

System.out.println("number is even ");

}else{

System.out.println("number is odd");

}

}

}

========================================================================

1. **Write a Java Program to find the Factorialof given number.**

import java.util.\*;

class Cdac{

public static int factorial(int num){

int fact=1;

if(num<0) System.out.println("Invalid");

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

fact = fact\*i;

}

return fact;

}

public static void main(String[] args) {

int n = factorial(5);

System.out.println(n);

}

}=========================================================================

1. **Find the Factorial of a number using Recursion.**

import java.util.\*;

class Cdac{

public static int fact(int num){

if(num==1) return 1;

return num\*fact(num-1);

}

public static void main(String[] args) {

int n = fact(4);

System.out.println(n);

}

}

=========================================================================

1. **Swap two numbers without using third variable approach 1.**

import java.util.\*;

class Cdac{

public static void swap\_1(int x,int y){

x = x+y;

y = x-y;

x = x-y;

System.out.println("X:"+x+" "+"Y:"+y);

}

public static void main(String[] args) {

swap\_1(10,20);

}

}

=======================================================================

1. **Swap two numbers without using third variable approach 2.**

import java.util.\*;

class Cdac{

public static void swap\_2(int x,int y){

x = x\*y;

y = x/y;

x = x/y;

System.out.println("X:"+x+" "+"Y:"+y);

}

public static void main(String[] args) {

swap\_2(10,20);

}

}

=======================================================================

1. **Swap two numbers without using third variable approach 3.**

import java.util.\*;

class Cdac{

public static void swap\_3(int x,int y){

x = x ^ y;

y = x ^ y;

x = x ^ y;

System.out.println("X:"+x+" "+"Y:"+y);

}

public static void main(String[] args) {

swap\_3(10,20);

}

}

=======================================================================

1. **How to check the given number is Positive or Negative in Java?**

import java.util.\*;

class Cdac{

public static void check(int num){

if(num>0) System.out.println(num+" "+": is positive");

else System.out.println(num+" "+": is neagative");

}

public static void main(String[] args) {

check(-2);

}

}

=======================================================================

1. **Write a Java Program to find whether given number is Leap year or NOT?**

import java.util.\*;

class Cdac{

public static void check\_leap(int num){

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

System.out.println(num+" "+": is leap year");

}

else {

System.out.println(num+" "+":is non leap year");

}

}

public static void main(String[] args) {

check\_leap(1900);

}

}

=======================================================================

1. **Write a Java Program to print the digits of a Given Number.**

import java.util.\*;

class Cdac {

public static void print\_digits(int num) {

int number = Math.abs(num); // Handle negative numbers

// Convert the number to a string to easily extract and print each digit

String numberStr = Integer.toString(number);

System.out.print("Digits are: ");

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

char digit = numberStr.charAt(i);

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

}

System.out.println();

}

public static void main(String[] args) {

print\_digits(-78);

}

}

=======================================================================

1. **Write a Java Program to print all the Factors of the Given number.**

import java.util.\*;

class Cdac{

public static int factorial(int num){

int fact=1;

if(num<0) System.out.println("Invalid");

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

fact = fact\*i;

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

}

return fact;

}

public static void main(String[] args) {

factorial(5);

}

}

=======================================================================

1. **Write a Java Program to find sum of the digits of a given number.**

import java.util.\*;

class Cdac{

public static int sum(int num){

if(num==0) return 0;

return num+sum(num-1);

}

public static void main(String[] args) {

int print = sum(5);

System.out.print("sum is :"+print);

}

}

=======================================================================

1. **Write a Java Program to find the smallest of 3 numbers(a,b,c) without using < or > symbol?**

import java.util.\*;

class Cdac{

public static void check\_smallest(int x,int y,int z){

int num = Math.min(x,Math.min(y,z));

System.out.println("minimum number is :"+num);

}

public static void main(String[] args) {

check\_smallest(50,15,20);

}

}

=======================================================================

1. **How to add two numbers without using the arithmetic operators in Java?**

import java.util.\*;

class Cdac{

public static int add(int x,int y){

// Iterate until there is no carry

while (y != 0) {

int carry = x & y; // Calculate carry bits

x = x ^ y; // Calculate sum without carry

y = carry << 1; // Shift carry bits to the left

}

return x;

}

public static void main(String[] args) {

int sum = add(50,20);

System.out.println(sum);

}

}

=======================================================================

1. **Write a java program to Reverse a given number.**

import java.util.\*;

class Cdac{

public static void reverse\_num(int x){

int temp = 0;

while (x != 0) {

int digit = x % 10; // Get the last digit

temp = temp \* 10 + digit; // Add digit to the reversed number

x = x / 10; // Remove the last digit from x

}

// Print the reversed number

System.out.println("Reversed number: " + temp);

}

public static void main(String[] args) {

reverse\_num(4321);

}

}

=======================================================================

1. **Write a Java Program to find GCD of two given numbers.**

import java.util.\*;

class Cdac {

public static int find\_GCD(int num1,int num2){

if(num2==0) return num1;

return find\_GCD(num2,num1%num2);

}

public static void main(String[] args) {

int n = find\_GCD(42,78);

System.out.println("GCD IS :"+" "+n);

}

}

=======================================================================

1. **Write a java program to LCM of TWO given number.**

import java.util.Scanner;

public class Cdac {

public static int find\_LCM(int a, int b) {

return (a \* b) / findGCD(a, b);

}

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

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

public static void main(String[] args) {

int lcm = find\_LCM(20,25);

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

}

}

=======================================================================

1. **Write a java program to LCM of TWO given number using Prime Factors method.**

import java.util.Scanner;

public class Cdac {

public static int find\_LCM(int a, int b) {

int max = Math.max(a, b);

int lcm = max;

while (lcm % a != 0 || lcm % b != 0) {

lcm += max;

}

return lcm;

}

public static void main(String[] args) {

int lcm = find\_LCM(25,20);

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

}

}

=======================================================================

1. **Check whether the Given Number is a Palindrome or NOT.**

import java.util.Scanner;

public class Cdac {

public static boolean Palindrome\_check(int num) {

int original\_Num = num;

int reversed\_Num = 0;

while (num != 0) {

int digit = num % 10;

reversed\_Num = reversed\_Num \* 10 + digit;

num /= 10;

}

if(original\_Num==reversed\_Num){

System.out.println("palindrome");

}else{

System.out.println("not palindrome");

}

return original\_Num == reversed\_Num;

}

public static void main(String[] args) {

Palindrome\_check(1231);

}

}

=======================================================================

1. **Write a Java Program to print all the Prime Factors of the Given Number.**

import java.util.Scanner;

public class Cdac {

public static void print\_factors(int num) {

if (num <= 1) {

System.out.println("No prime factors for numbers less than or equal to 1.");

return;

}

// Check for number of 2s that divide num

while (num % 2 == 0) {

System.out.print(2 + " ");

num /= 2;

}

// Now check for odd numbers from 3 onwards

for (int i = 3; i <= num; i += 2) {

while (num % i == 0) {

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

num /= i;

}

}

// If num is still greater than 2, then num itself is a prime number

if (num > 2) {

System.out.print(num);

}

}

public static void main(String[] args) {

print\_factors(58);

}

}

=======================================================================

1. **Write a Java Program to check whether the Given Number is Prime Number or NOT.**

import java.util.Scanner;

public class Cdac {

public static boolean isPrime(int num) {

if (num <= 1) {

return false;

}

if (num <= 3) {

return true; // 2 and 3 are prime numbers

}

// Eliminate multiples of 2 and 3

if (num % 2 == 0 || num % 3 == 0) {

return false;

}

// Check for factors from 5 to sqrt(num)

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

if (num % i == 0 || num % (i + 2) == 0) {

return false;

}

}

return true;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int num = scanner.nextInt();

if (isPrime(num)) {

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

} else {

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

}

}

}

=======================================================================

1. **Write a Java Program to print Prime Numbers from 1 to N.**

import java.util.Scanner;

public class Cdac {

public static boolean isPrime(int num) {

if (num <= 1) {

return false;

}

if (num <= 3) {

return true; // 2 and 3 are prime numbers

}

// Eliminate multiples of 2 and 3

if (num % 2 == 0 || num % 3 == 0) {

return false;

}

// Check for factors from 5 to sqrt(num)

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

if (num % i == 0 || num % (i + 2) == 0) {

return false;

}

}

return true;

}

public static void print\_Prime(int N) {

if (N < 2) {

System.out.println("No prime numbers less than 2.");

return;

}

System.out.println("Prime numbers from 1 to " + N + " are:");

for (int num = 2; num <= N; num++) {

if (isPrime(num)) {

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

}

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int N = scanner.nextInt();

print\_Prime(N);

}

}

=======================================================================

1. **Write a Java Program to check whether the given number is Armstrong Number or NOT.**

import java.util.Scanner;

public class Cdac {

public static boolean isArmstrong(int number) {

int originalNumber = number;

int sum = 0;

int digits = String.valueOf(number).length();

while (number > 0) {

int digit = number % 10;

sum += Math.pow(digit, digits);

number /= 10;

}

return sum == originalNumber;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (isArmstrong(number)) {

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

} else {

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

}

}

}

=======================================================================

1. **Write a Java Program to print Armstrong Number is between 1 to 1000.**

public class Cdac {

public static boolean isArmstrong(int number) {

int originalNumber = number;

int sum = 0;

int digits = String.valueOf(number).length();

while (number > 0) {

int digit = number % 10;

sum += Math.pow(digit, digits);

number /= 10;

}

return sum == originalNumber;

}

public static void main(String[] args) {

System.out.println("Armstrong numbers between 1 and 1000 are:");

for (int i = 1; i <= 1000; i++) {

if (isArmstrong(i)) {

System.out.println(i);

}

}

}

}

=====================================================================

1. **Write a Java Program to check whether the given number is Perfect Number or NOT.**

import java.util.Scanner;

public class Cdac {

public static boolean isPerfect(int number) {

int sum = 0;

for (int i = 1; i <= number / 2; i++) {

if (number % i == 0) {

sum += i;

}

}

return sum == number;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int number = scanner.nextInt();

if (isPerfect(number)) {

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

} else {

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

}

}

}

=====================================================================

1. **Write a Java Program to print Perfect Numbers between 1 to 1000.**

public class Cdac {

public static boolean isPerfect(int number) {

int sum = 0;

for (int i = 1; i <= number / 2; i++) {

if (number % i == 0) {

sum += i;

}

}

return sum == number;

}

public static void main(String[] args) {

System.out.println("Perfect numbers between 1 and 1000 are:");

for (int i = 1; i <= 1000; i++) {

if (isPerfect(i)) {

System.out.println(i);

}

}

}

}