**Assignment 1**

**Solve the assignment with following thing to be added in each question.**

-Program

-Flow chart

-Explanation

-Output

-Time and Space complexity

1. Armstrong Number

Problem: Write a Java program to check if a given number is an Armstrong number.

Test Cases:

Input: 153

Output: true

Input: 123

Output: false

import java.util.Scanner;

public class Armstrong {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

int originalNum = num; // Store the original number

int temp, total = 0;

// Calculate the sum of cubes of each digit

while (num != 0) {

temp = num % 10;

total = total + temp \* temp \* temp;

num = num/10;

}

// Check if the original number is equal to the total

if (total == originalNum) {

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

} else {

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

}

sc.close();

}

}

-Explanation- In the Armstrong program I use while loop for finding the cube of digit.

while (num != 0) {

temp = num % 10;

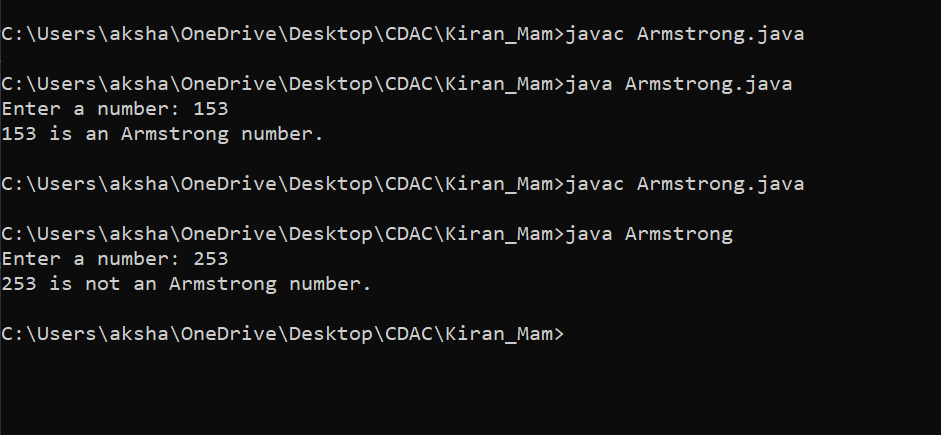
total = total + temp \* temp \* temp;

num = num/10;

**while syntax-while(condition){**

**// loop code}**

**OUTPUT**



2. Prime Number

Problem: Write a Java program to check if a given number is prime.

Test Cases:

Input: 29

Output: true

Input: 15

Output: false

import java.util.Scanner;

public class Prime {

public static boolean checkPrimeFunc(int nums) {

if (nums <= 1) {

return false;

}

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

if (nums % i == 0) {

return false;

}

}

return true;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

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

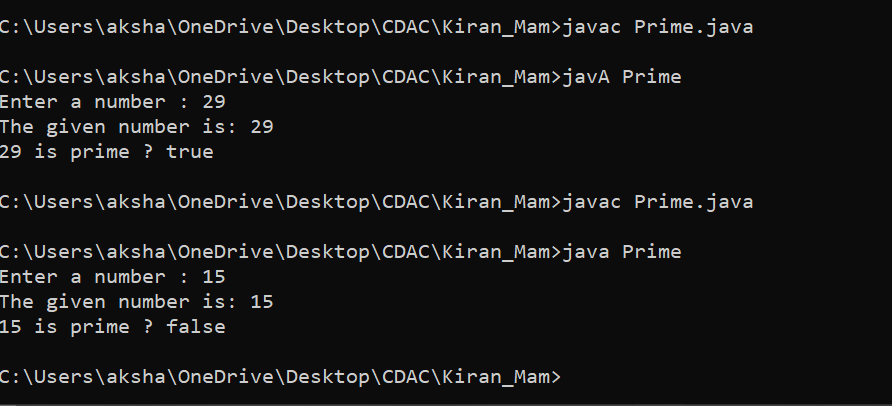
System.out.println(num + " is prime ? " + checkPrimeFunc(num));

}

}

-Explanation

**OUTPUT**



3. Factorial

Problem: Write a Java program to compute the factorial of a given number.

Test Cases:

Input: 5

Output: 120

Input: 0

Output: 1

import java.util.Scanner;

public class Factorial

{

public static void main(String []args)

{

Scanner sc=new Scanner(System.in);

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

int num=sc.nextInt();

int i=1,fact=1;

while(i<=num)

{

fact=fact\*i;

i++;

}

System.out.println("Factorial of the number: "+fact);

}

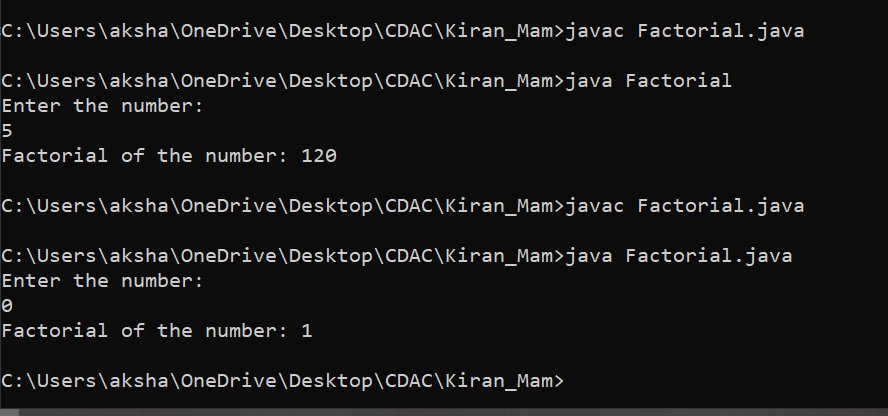
}

-Explanation-In the Factorial program first intialise I and fact by 1 then using while loop for finding the fact of given number.

**while syntax-while(condition){**

**// loop code}**

**OUTPUT**



4. Fibonacci Series

Problem: Write a Java program to print the first n numbers in the Fibonacci series.

Test Cases:

Input: n = 5

Output: [0, 1, 1, 2, 3]

Input: n = 8

Output: [0, 1, 1, 2, 3, 5, 8, 13]

import java.util.Scanner;

public class Fibonacci {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

int a=0,b=1;

System.out.println("The fibonacci series is ");

System.out.println(a+" ");

System.out.println(b+" ");

for(int i=3; i<=n;i++){

int c=a+b;

System.out.println(c+" ");

a=b;

b=c;

}

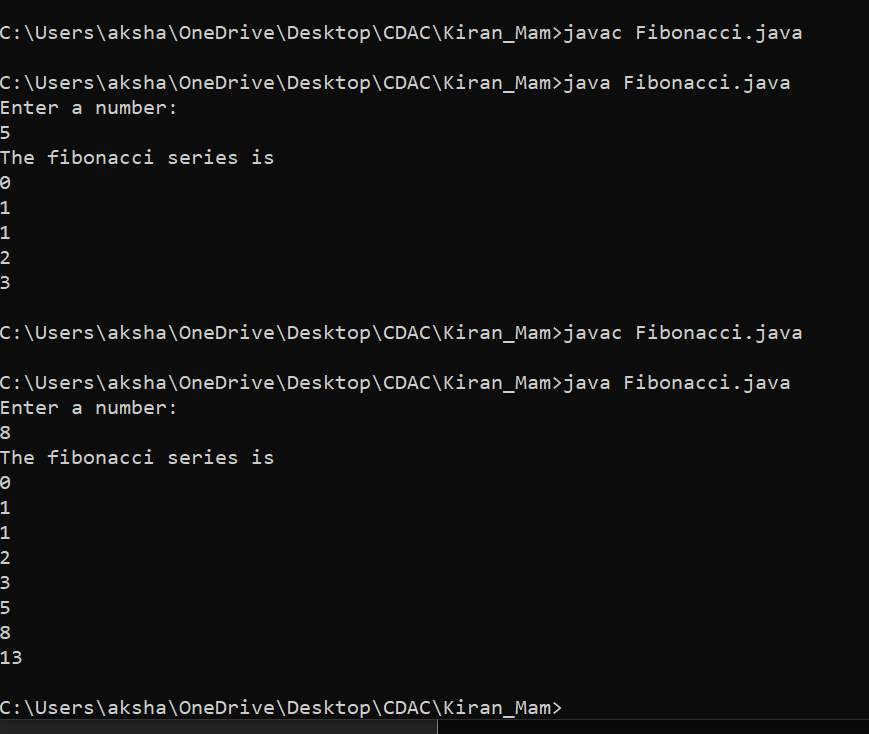
}

}

-Explanation -In the Fibonacci series program initialize the a=0,b=1 using for loop for Fibonacci series.

For loop syntax- For(initialization;condition;updation)

**OUTPUT**



5. Find GCD

Problem: Write a Java program to find the Greatest Common Divisor (GCD) of two numbers.

Test Cases:

Input: a = 54, b = 24

Output: 6

Input: a = 17, b = 13

Output: 1

import java.util.Scanner;

public class GCD

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

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

int x = sc.nextInt();

int y = sc.nextInt();

System.out.println("The GCD of two numbers is: " + findGCD(x,y));

}

static int findGCD(int x, int y)

{

int r=0, a, b;

a = (x > y) ? x : y; // a is greater number

b = (x < y) ? x : y; // b is smaller number

r = b;

while(a % b != 0)

{

r = a % b;

a = b;

b = r;

}

return r;

}

}

-Explanation- In the GCD program logic-

while(a % b != 0)

{

r = a % b;

a = b;

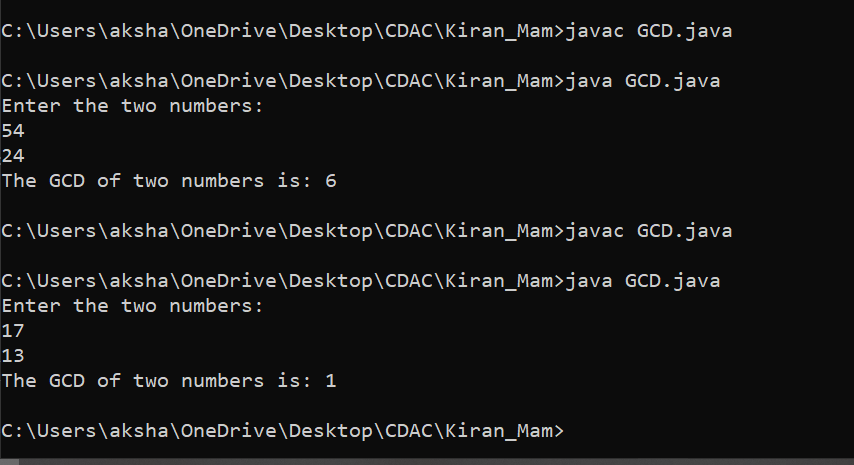
b = r;

}

**while syntax-while(condition){**

**// loop code}**

**OUTPUT**



6. Find Square Root

Problem: Write a Java program to find the square root of a given number (using integer approximation).

Test Cases:

Input: x = 16

Output: 4

Input: x = 27

Output: 5

import java.util.Scanner;

public class Squareroot{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

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

double number = sc.nextDouble();

double squareRoot = Math.sqrt(number);

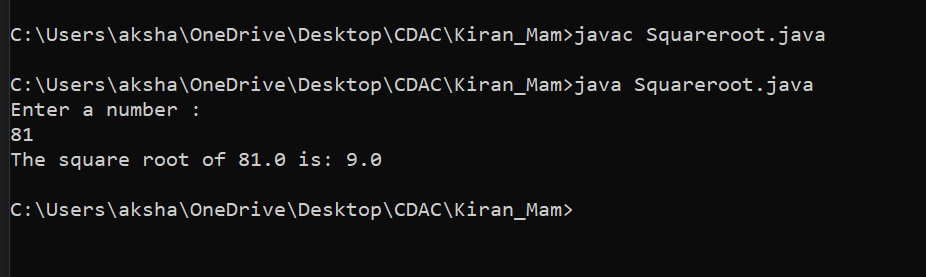
System.out.println("The square root of " + number + " is: " + squareRoot);

}

}

-Explanation- The square root program the square root finding using Math function squrt(number)

**OUTPUT**



7. Integer Palindrome

Problem: Write a Java program to check if a given integer is a palindrome.

Test Cases:

Input: 121

Output: true

Input: -121

Output: false

import java.util.Scanner;

class Palindrome {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int num = sc.nextInt();

int num1 = num; // Store the original number

int reverse = 0;

while (num != 0) {

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

reverse = reverse \* 10 + last; // Update the reverse number

num = num / 10; // Remove the last digit

}

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

if (num1 == reverse) {

System.out.println("It is a palindrome number.");

} else {

System.out.println("It is not a palindrome number.");

}

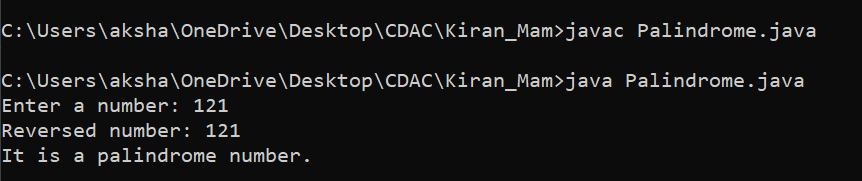
sc.close();

}

}

-Explanation- The palindrome program the finding using while and if loop we can find out given number is palindrome or not.

**OUTPUT**



8. Leap Year

Problem: Write a Java program to check if a given year is a leap year.

Test Cases:

Input: 2020

Output: true

Input: 1900

Output: false

import java.util.Scanner;

class LeapYear {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

// Use the scanner instance to read the integer

int n = scanner.nextInt();

// Check if the year is a leap year

if (n % 4 == 0) {

System.out.println("The entered year is a leap year.");

} else {

System.out.println("The entered year is not a leap year.");

}

scanner.close();

}

}

-Explanation- Using if loop condition we can find out given year leap or not.

OUTPUT

