**C-DAC Mumbai Date 25/09/2024**

**Subject: Algorithm and Data Structure**

**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

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

class PrimeNo{

static boolean prime(int n){

if(n<=1)

return false;

if(n % 2 !=0)

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

{

if (n % i == 0)

{

return false;

}

}

return true;

}

public static void main(String args[]){

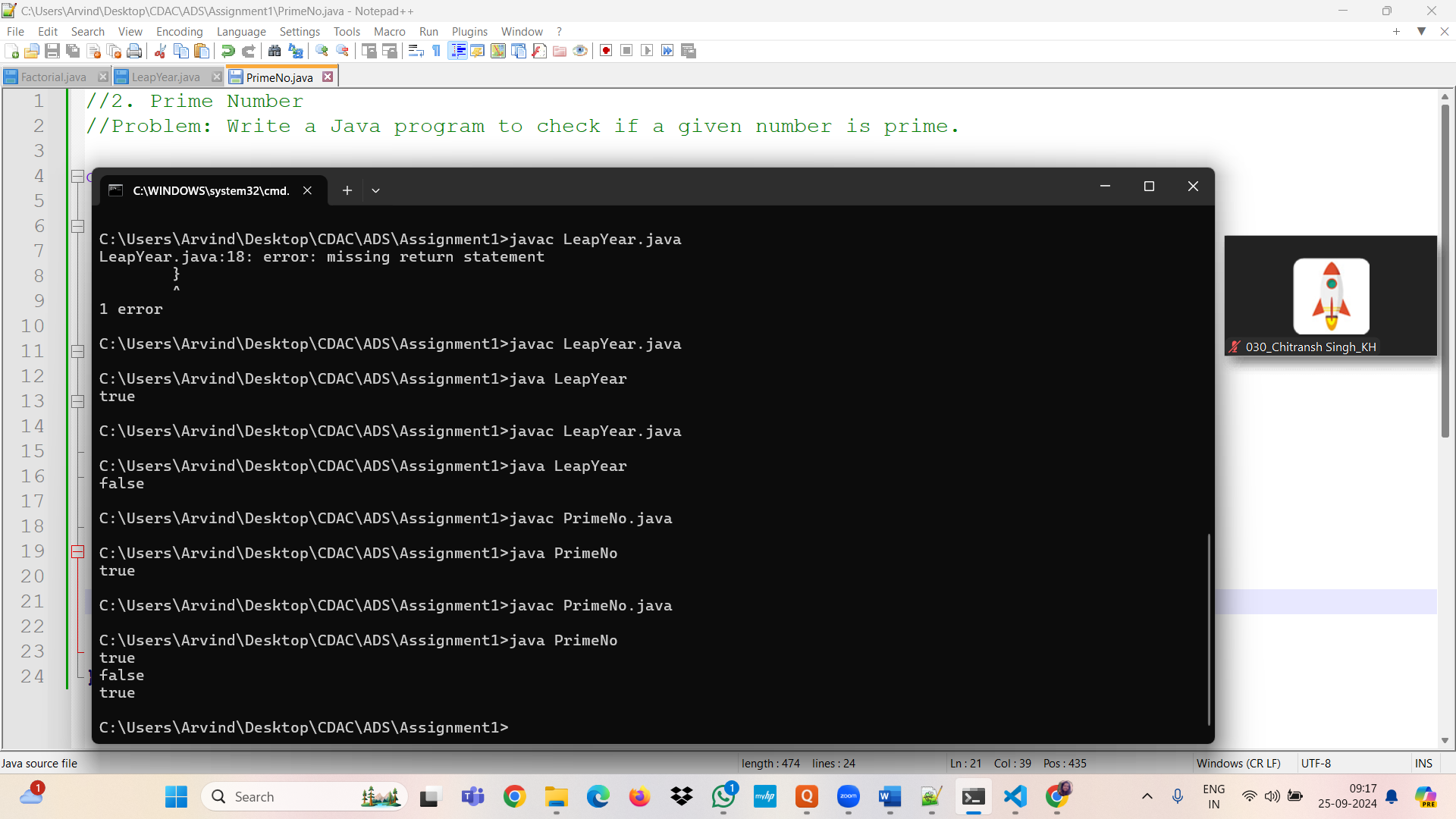
System.out.println(prime(29));

System.out.println(prime(15));

System.out.println(prime(2));

}

}



3. Factorial

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

class Factorial{

static int fact(int n)

{

if(n<=1){

return 1;

}

else

{

return n \*fact(n-1);

}

}

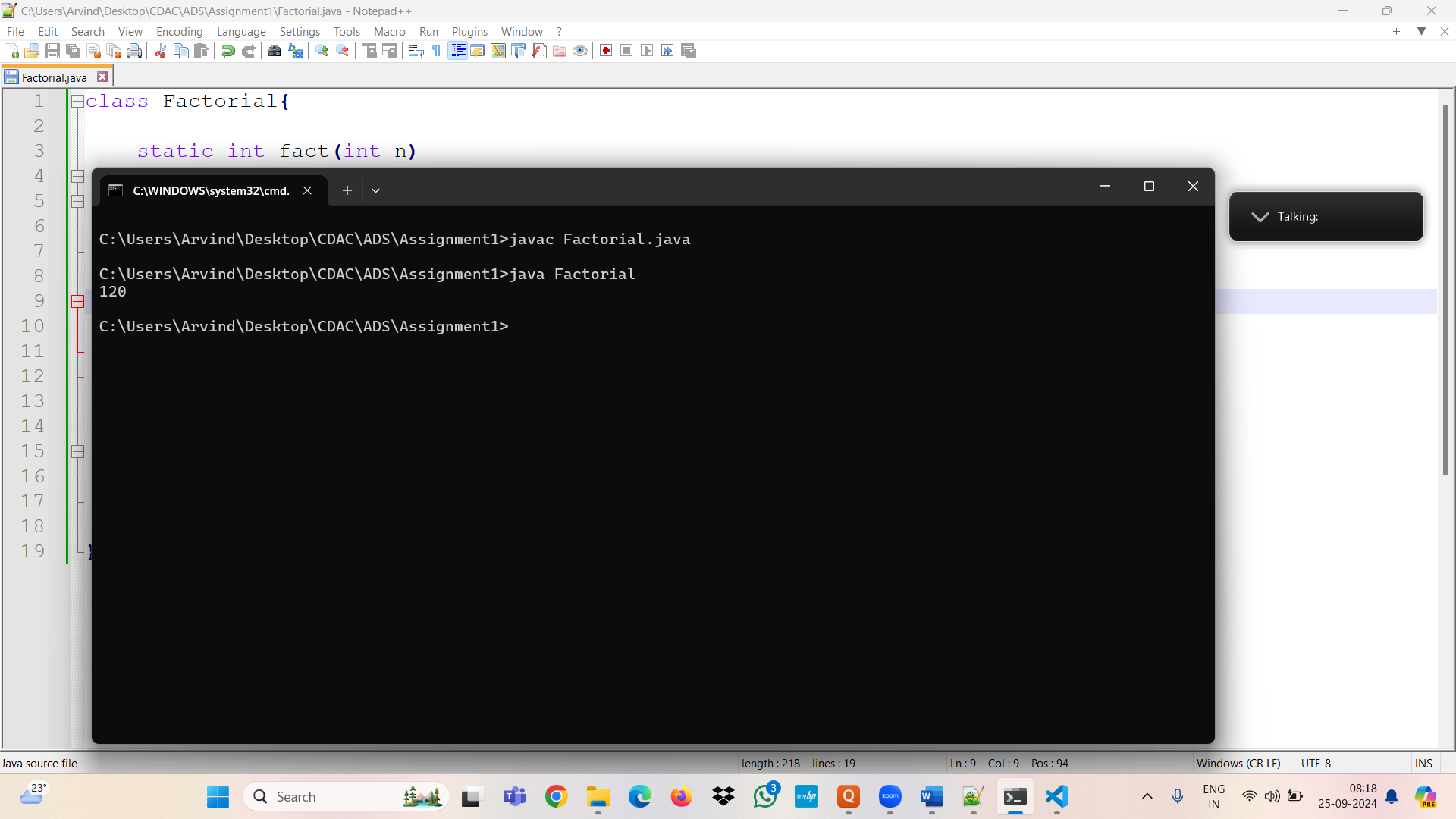
public static void main(String args[])

{

System.out.println(fact(5));

}

}



Test Cases:

Input: 5

Output: 120

Input: 0

Output: 1

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]

class Fibonacci{

static int fib(int n){

if(n <= 1)

{

return n;

}

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

}

public static void main(String args[]){

int num = 10;

for(nt i =1; i<=num; i++)

{

System.out.println((fib(i)+ " "));

}

}

//when num = 100 ; the integer value reaches its limit and its start to go weird and print negative.

}

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

6. Find Square Root

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

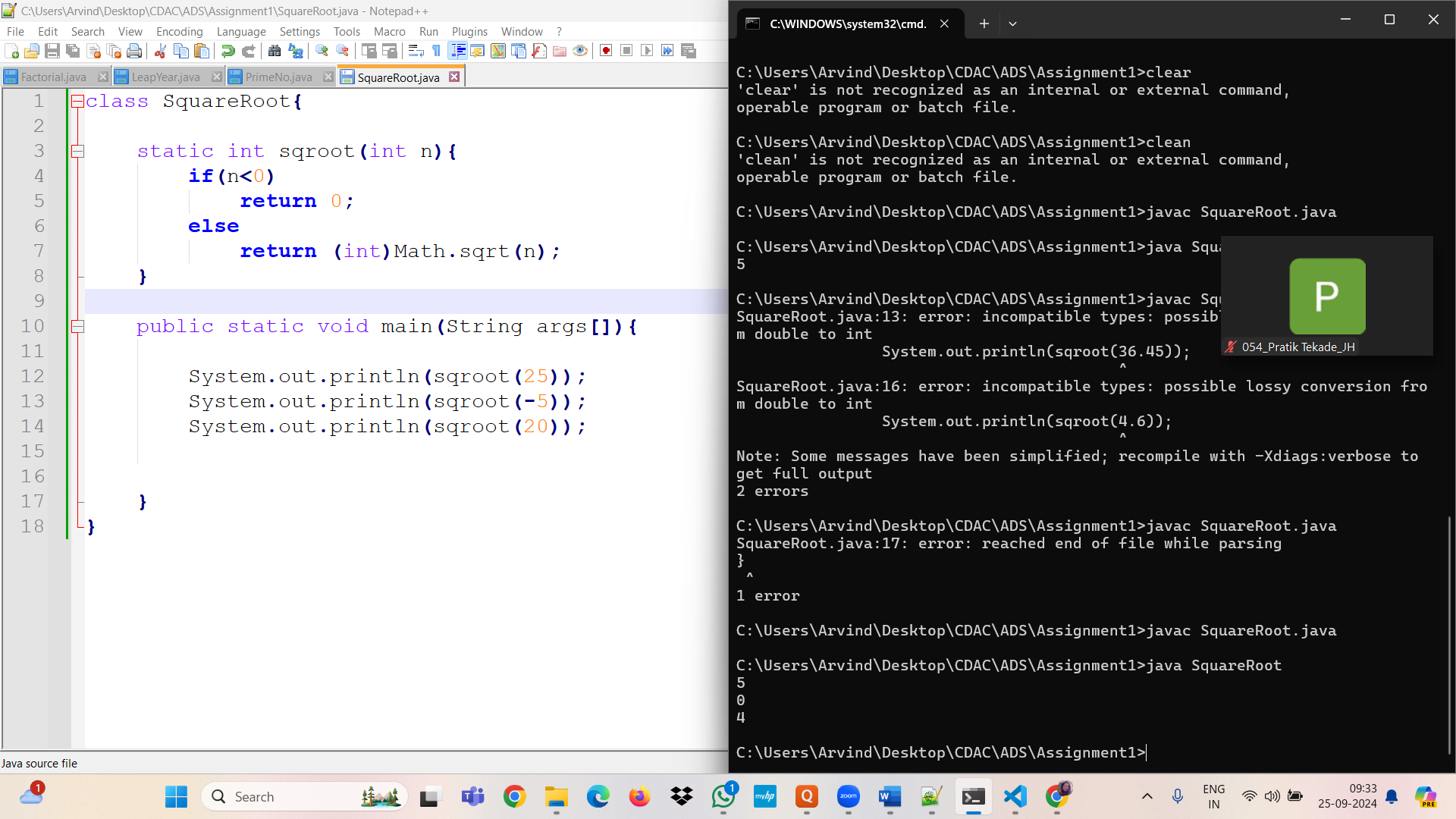
Test Cases:

Input: x = 16

Output: 4

Input: x = 27

Output: 5



7. Find Repeated Characters in a String

Problem: Write a Java program to find all repeated characters in a string.

Test Cases:

Input: "programming"

Output: ['r', 'g', 'm']

Input: "hello"

Output: ['l']

8. First Non-Repeated Character

Problem: Write a Java program to find the first non-repeated character in a string.

Test Cases:

Input: "stress"

Output: 't'

Input: "aabbcc"

Output: null

9. 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

10. 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

class LeapYear{

static boolean year(int n){

if(n<0){

return false;

}

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

return true;

}

return false;

}

public static void main(String args[]){

System.out.println(year(1900));

}

}

