**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.

import java.util.Scanner;

class Q1ArmStr{

public static void main(String[] args){

Scanner sc = new Scanner(System.in); //Scanner file to take input(s)

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

int num = sc.nextInt(); //Taking i/p from user, storing in var

int temp = num; //var to operate on i/p

int armStr = 0; //var to store o/p

int digit=0; //var to count no of digits

while(temp!=0){

digit++; //incrementing digit count

temp = temp/10; //discarding last digit of temp

}

temp = num;

while(temp!=0){ //base condition

armStr += Math.pow(temp%10,digit); //adding n'th power of last digit of temp

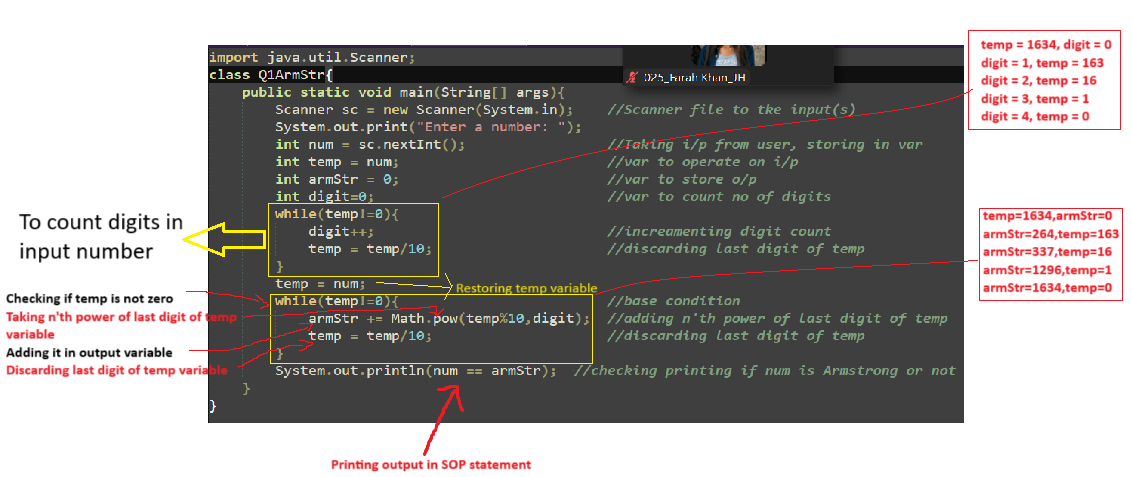
temp = temp/10; //discarding last digit of temp

}

System.out.println(num == armStr); //checking printing if num is Armstrong or not

}

}



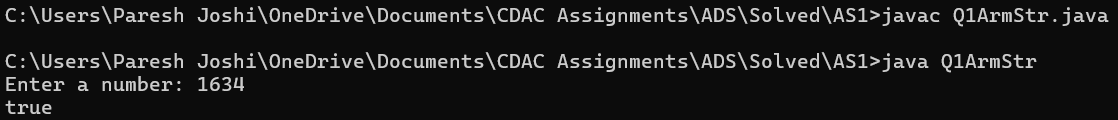
Test Cases:

Input: 153

Output: true

Input: 123

Output: false



2. Prime Number

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

import java.util.Scanner;

class Q2PrimeNum{

private static String isPrime(int num){

for(int i=2; i<=num/2;i++){ //for loop till half of number

if(num%i==0){ //Checking if num is divisible by any iterating number

return " is a prime number"; //if true return positive String

}

}

return " is not a prime number"; //else retun negative String

}

public static void main(String[] args){

Scanner sc = new Scanner(System.in); //Scanner file to tke input(s)

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

int num = sc.nextInt(); //Taking i/p from user, storing in var

System.out.println("numer " + num + isPrime(num)); //calling function to direct into SOP statement

}

}

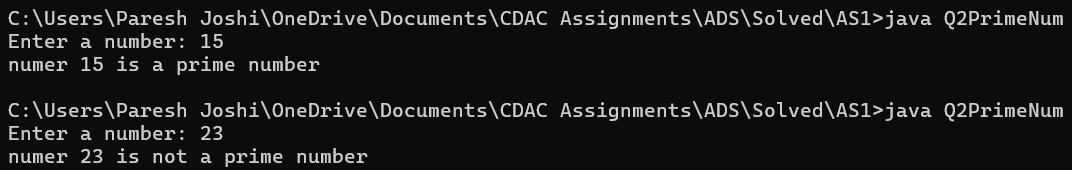
Test Cases:

Input: 29

Output: true

Input: 15

Output: false



3. Factorial

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

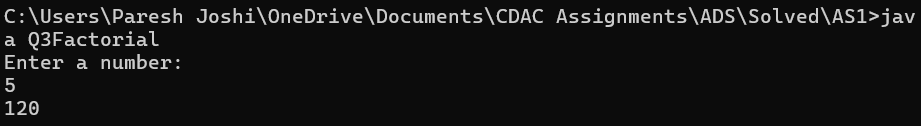
Test Cases:

Input: 5

Output: 120

Input: 0

Output: 1



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]

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.

import java.util.Scanner;

class Q8NonRepeated{

private static char isNotRepeated(String str){

char c;

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

boolean bool=true;

for(int j=0; j<str.length();j++){

if(j!=i && str.charAt(i)==str.charAt(j)){

bool = false;

break;

}

}

if(bool)

return str.charAt(i);

}

return '-';

}

public static void main(String[] args){

Scanner sc = new Scanner(System.in); //Scanner file to tke input(s)

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

String str = sc.nextLine(); //Storing i/p from user in var

System.out.println(isNotRepeated(str));

}

}

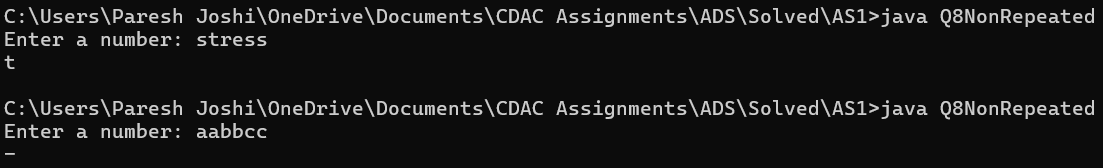
Test Cases:

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.

import java.util.Scanner;

class Q9Palindrom{

private static boolean isPalindrom(int num){

int digit;

int revNum=0;

int temp = num;

while(temp!=0){ //checking till number != zero

digit = temp%10; //Storing last digit of number

revNum = revNum\*10 + digit;

//Multiply revNum by 10 to get decimal space then adding fetched digit into reverse number

temp = temp/10; //Discarding last digit of number

}

return num==revNum; //returning boolean value

}

public static void main(String[] args){

Scanner sc = new Scanner(System.in); //Scanner file to tke input(s)

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

int num = sc.nextInt(); //Storing i/p from user in var

String pal = isPalindrom(num) ? " Number is Palindrom":" Number is not Palindrom"; //ternary operator to define o/p

System.out.println(num + pal);

}

}

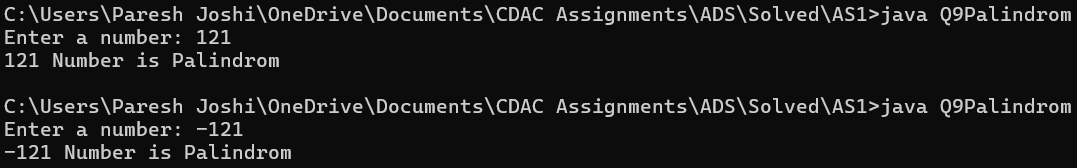
Test Cases:

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.

import java.util.Scanner;

class Q10LeapYear{

private static void isLeapYr(int num){

if(num%4==0 && (num%100!=0 || num%400==0)) //if condition is true

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

else //else

System.out.println(num + " is not leap year"); //year is not leap year

}

public static void main(String[] args){

Scanner sc = new Scanner(System.in); //Scanner file to tke input(s)

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

int num = sc.nextInt(); //Storing i/p from user in var

isLeapYr(num);

}

}

Test Cases:

Input: 2020

Output: true

Input: 1900

Output: false

