**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

Ans = import java.util.\*;

import java.lang.\*;

class Main {

public static void main(String[] args)

{

int num = 1634, reverse = 0;

int len = order(num);

if (num == getArmstrongSum(num, len))

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

else

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

}

private static int getArmstrongSum(int num, int order) {

if(num == 0)

return 0;

int digit = num % 10;

return (int) Math.pow(digit, order) + getArmstrongSum(num/10, order);

}

private static int order(int num) {

int len = 0;

while (num!=0)

{

len++;

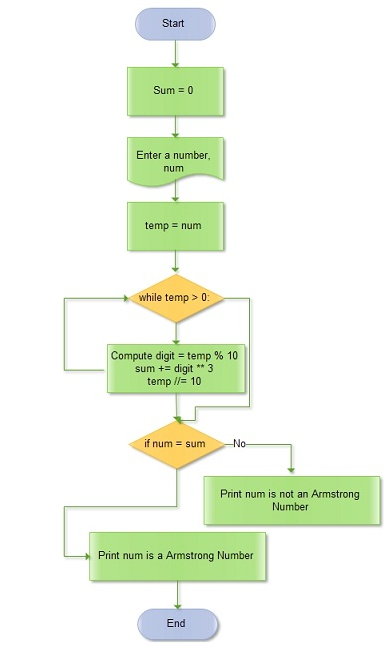
num = num/10;

}

return len;

}

}



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

Ans = class Program{

static boolean printFun(int n, int i){

if (n<=2)

return (n==2) ? true:false;

if (n%i ==0)

return false;

if (i\*i > n)

return true;

return printFun(n,i+1);

}

public static void main(String args[]){

if (printFun(39,8))

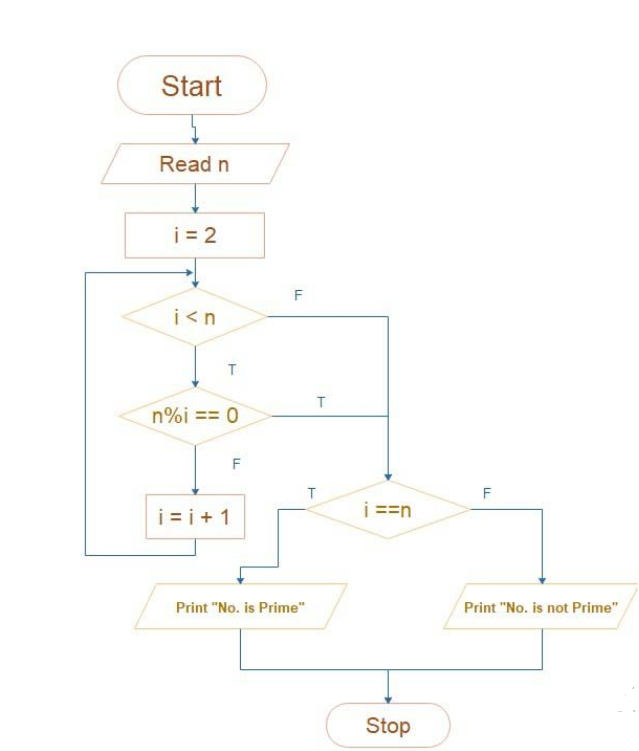
System.out.println("Yes, it's a prime number!");

else

System.out.println("No, it's not a prime number!");

}

}



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

Ans = class Program{

static int printFun(int n){

if (n==0)

return 1;

return n\*printFun(n-1);

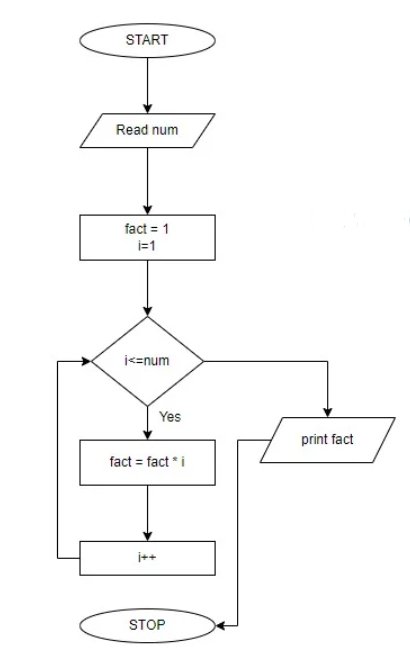
}

public static void main(String args[]){

System.out.println(printFun(4));

}

}



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]

Ans = class Program{

static int n=0 , n1=1, n2= 0;

static void printFun(int c){

if (c>0){

n2 = n+n1;

n = n1;

n1 = n2;

System.out.print(" "+n2);

printFun(c-1);

}

}

public static void main(String args[]){

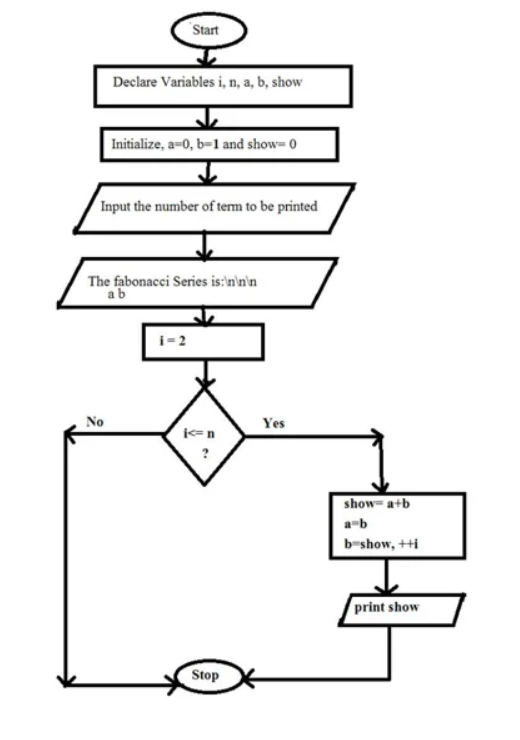
int c =10;

System.out.print(n+" "+n1);

printFun(c-2);

}

}



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

Ans = public class Solution {

int a = 25;

int b = 40;

int c = 1;

public void prog(){

for (int i = 1; i <=a && i<=b; i++) {

if (a%i == 0 && b%i==0) {

c = i;

}

}

System.***out***.println("GCD is "+c);

}

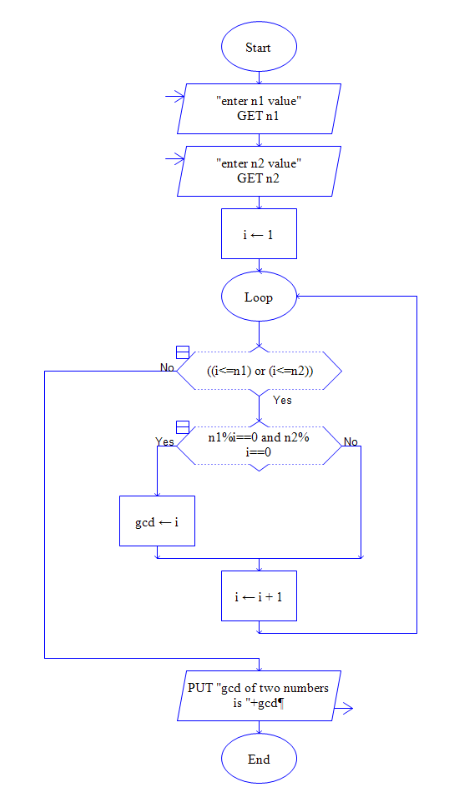
public static void main(String[] args) {

Solution solution = new Solution();

solution.prog();

}

}



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

Ans = public class Solution {

static int prog(int x) {

if (x == 0 || x == 1)

return x;

int i = 1, result = 1;

while(result<=x) {

i++;

result = i\*i;

}

return i-1;

}

public static void main(String[] args) {

System.***out***.println(*prog*(24));

}

}

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']

Ans= package prac;

import java.util.Arrays;

public class Solution {

static void prog(String str) {

System.***out***.println("The string is: " + str);

char[] carray = str.toCharArray();

Arrays.*sort*(carray);

System.***out***.println("Duplicate Characters in the above string are: ");

for (int i = 0; i < carray.length - 1; i++) {

if (carray[i] == carray[i + 1]) {

System.***out***.println(carray[i]);

while (i < carray.length - 1 && carray[i] == carray[i + 1]) {

i++;

}

}

}

}

public static void main(String[] args) {

*prog*("programming");

}

}

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

Ans = **class** Main {

**public** **static** **void** main(**String**[] args) {

**String** string = "akash";

**int** index = -1;

**char** fnc = ' ';

**for** (**char** i : string.toCharArray()) {

**if** (string.indexOf(i) == string.lastIndexOf(i)) {

fnc = i;

**break**;

}

**else** {

index += 1;

}

}

**if** (index == 1) {

System.out.println("Either all characters are repeating or "

+ "string is empty");

}

**else** {

System.out.println("First non-repeating character is " + fnc);

}

}

}

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

Ans = public class Solution2 {

public static void main(String[] args) {

int num = 454;

int sum = 0;

int r,temp;

temp=num;

while(num>0) {

r = num%10;

sum = (sum\*10)+r;

num = num/10;

}

if(temp==sum)

System.***out***.println("Palindrome");

else

System.***out***.println("Not Palindrome!");

}

}

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

Ans = public class Main {

public static void main(String[] args) {

int year = 1900;

boolean leap = false;

if (year % 4 == 0) {

if (year % 100 == 0) {

if (year % 400 == 0)

leap = true;

else

leap = false;

}

else

leap = true;

}

else

leap = false;

if (leap)

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

else

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

}

}

Some qyestions by Ma’am :

Q- ***Palindrome***

Ans = package prac;

public class Solution2 {

public static void main(String[] args) {

int num = 454;

int sum = 0;

int r,temp;

temp=num;

while(num>0) {

r = num%10;

sum = (sum\*10)+r;

num = num/10;

}

if(temp==sum)

System.***out***.println("Palindrome");

else

System.***out***.println("Not Palindrome!");

}

}

Q- ***Remove punctuations?***

Ans = package prac;

public class Solution1 {

public static void main(String[] args) {

String str = "%Welcome to @cdacmumbai?<";

str = str.replaceAll("\\p{Punct}", "");

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

}

}