**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.

Solution—

**package** assign1;

**import** java.util.Scanner;

**public** **class** A1\_Q1 {

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

Scanner sc=**new** Scanner(System.***in***);

**int** n=sc.nextInt();

**int** temp=n;

**int** sum=0;

**while**(n>0) {

**int** rem=n%10;

sum=sum+rem\*rem\*rem;

n=n/10;

}

**if**(sum==temp) {

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

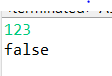
}

**else**

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

}

}



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.

Solution—

**package** assign1;

**import** java.util.Scanner;

**public** **class** A1\_Q2 {

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

Scanner sc=**new** Scanner(System.***in***);

**int** n=sc.nextInt();

**int** count=0;

**if**(n<2)

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

**else** {

**for**(**int** i=2;i<n;i++) {

**if**(n%i==0)

count++;

}

**if**(count>1)

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

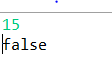
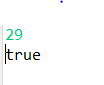
**else**

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

}

}

}



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.

Solution—

**package** assign1;

**import** java.util.Scanner;

**public** **class** A1\_Q3 {

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

Scanner sc=**new** Scanner(System.***in***);

**int** n=sc.nextInt();

**int** fact=1;

**while**(n>0) {

fact=fact\*n;

--n;

}

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

}

}



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.

**package** assign1;

**import** java.util.Scanner;

**public** **class** A1\_Q4 {

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

Scanner sc=**new** Scanner(System.***in***);

**int** n=sc.nextInt();

**int** f1=0,f2=1;

**int** f=0;

**for**(**int** i=1;i<=n;i++) {

System.***out***.print(f);

**if**(i<n)

System.***out***.print(",");

f1=f;

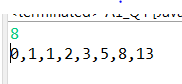
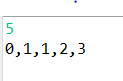
f=f2;

f2=f2+f1;

}

}

}



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).

Solution—

**package** assign1;

**import** java.util.Scanner;

**public** **class** A1\_Q6 {

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

Scanner sc=**new** Scanner(System.***in***);

**int** n=sc.nextInt();

**for**(**int** i=1;i<=n;i++) {

**if**(i\*i==n) {

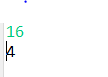
System.***out***.println(i);

}

}

}

}



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.

Solution—

**import** java.util.Scanner;

**public** **class** A1\_Q7 {

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

Scanner sc=**new** Scanner(System.***in***);

String str=sc.nextLine();

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

**int** count=0;

**for**(**int** j=i+1;j<str.length();j++) {

**if**(str.charAt(j)==str.charAt(i))

count++;

}

**if**(count>0) {

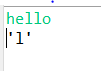
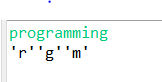
System.***out***.print("'"+str.charAt(i)+"'");

}

}

}

}



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.

**Solution—**

**package** assign1;

**import** java.util.Scanner;

**public** **class** A1\_Q10 {

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

Scanner sc=**new** Scanner(System.***in***);

**int** n=sc.nextInt();

**int** temp= n;

**int** sum=0;

**if**(n>0) {

**while**(n>0) {

**int** rem=n%10;

sum=sum\*10+rem;

n=n/10;

}

**if**(sum==temp)

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

**else**

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

}

**else** {

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

}

}

}



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.

Solution—

**package** assign1;

**import** java.util.Scanner;

**public** **class** A1\_Q11 {

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

Scanner sc=**new** Scanner(System.***in***);

**int** year=sc.nextInt();

**boolean** leap = **false**;

**if** (year % 4 == 0) {

**if** (year % 100 == 0) {

**if** (year % 400 == 0)

leap = **true**;

**else**

leap = **false**;

}

**else**

leap = **true**;

}

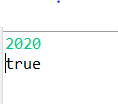
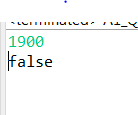
**else**

leap = **false**;

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

}

}

Test Cases:

Input: 2020

Output: true

Input: 1900

Output: false