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

```
package org.example;
import java.util.Scanner;

public class ArmstrongNumber {

            public static boolean isArmstrong(int num){
              int original  = num ;
              int sum = 0 ;
              int n = String.valueOf(num).length();

              while (num != 0) {
                int digit = num % 10;
                sum += Math.pow(digit,n);
                num /= 10;
              }
              return sum == original;
            }

           public static void main(String[] args) {
             Scanner sc =new Scanner(System.in);
             System.out.println("Enter a number : " );
             int num = sc.nextInt();
             System.out.println(isArmstrong(num));
              sc.close();
}
}
```

Flowchart:

```
1.Start
2.Input the number
3.Store the number in a variable (original)
4. Count number of digits
5. Initialize sum to 0
6. For each digit of given number :
    > Extract the last digit
    > Raise the digit to the power of no of digits
    > Add result to (sum)
    > Remove the last digit from the number
7.If sum is equal to original,return TRUE otherwise return FALSE
8.End
```

Time Complexity :

>> O(d),where d is no of digits in the input number.
The program extracts each digit,raises it to the power of d , and sums the result.

Space Complexity :

>> O(1) : only few integer variables are used to store the result

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

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.

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

1)Program

```
package org.example;

import java.util.Scanner;

public class LeapYearChecker {
        public static void main(String[] args){
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter a year: ");
            int year = scanner.nextInt();
            if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
                System.out.println("true");
            } else {
                System.out.println("false");
            }
        }
    }
```

Time Complexity : O(1)
Space Complexity : O(1)

Flow Chart-

Start: Begin the algorithm.

Input Year: Prompt the user to enter a year and read the input value into the variable year.

Check Conditions:

Condition 1: Check if year is divisible by 4. And not divisible by 100.

If YES,proceed to the next condition

If NO,print False.

Condition 2: Check if year is divisible by 400.

If YES,print True

If No,print False

End:The algorithm end.

OutPut-

```
Enter a year: 2020

True

Enter a year: 1900

false
```