C-DAC Mumbai

Date 25/09/2024

Subject: Algorithm and Data Structure
Assignment 1

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.

```java
import java.util.*;
class ArmStrongNumber{

    public static boolean isArmstrong(int number){
        int original_no = number;
        int sum = 0;
        int numberOfDigits = String.valueOf(number).length();


        while(number>0){
            int digit = number%10;
            sum += Math.pow(digit, numberOfDigits);
            number = number/10;
        }
        return sum == original_no;
    }


    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a number : ");
        int number = sc.nextInt();

        if(isArmstrong(number)){
            System.out.println(number + " is armstrong number.");
        }
        else{
            System.out.println(number + " is not an armstrong number.");
        }

        sc.close();
    }

}
```

```
C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>javac q1.java

C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java ArmStrongNumber
Enter a number :
121
121 is not an armstrong number.

C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java ArmStrongNumber
Enter a number :
151
151 is not an armstrong number.
```

Time Complexity: O(log10(n))
Space Complexity: O(1)

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.

```java
import java.util.*;
class PrimeOrNotPrime{

    public static boolean isPrime(int number){

        if(number <= 1){
            return false;
        }

        else if(number == 2){
            return true;
        }
        else if(number%2 == 0){
            return false;
        }
        else{
            return true;
        }

    }


    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number : ");
        int number = sc.nextInt();

        if(isPrime(number)){
            System.out.println(number + " is a prime number.");
        }
        else{
            System.out.println(number + " is not a prime number.");
        }

        sc.close();
    }
}
```

```
C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java PrimeOrNotPrime
Enter the number :
23
23 is a prime number.

C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java PrimeOrNotPrime
Enter the number :
12
12 is not a prime number.
```

Time complexity → O(1)
Space complexity → O(1)

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.

```java
import java.util.*;


class factorial{

    static int fact(int num){
        if(num<=1){
            return 1;
        }
        else{
            return num*fact(num-1);
        }
    }


    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number : ");
        int num = sc.nextInt();
        int factorial  = fact(num);
        System.out.println("the factorial of " + num + " is " +  factorial);
    }
}
```

```
C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java factorial
Enter the number :
5
the factorial of 5 is 120
```

Time Complexity → O(n)
Space Complexity → O(n)

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.

```java
import java.util.Scanner;

class fibonacci{

    static int fib(int n){
        if(n<=1){
            return n;
        }
        else{
            return fib(n-1)+fib(n-2);
        }

    }


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

        for(int i = 0; i<= num; i++ )
        {
            System.out.print(fib(i) + " " );
        }
    }
}
```

```
C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java fibonacci
Enter the number :
10
0 1 1 2 3 5 8 13 21 34 55
```

Time Complexity → $O(2^n)$
Space Complexity → $O(n)$

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.

```java
import java.util.*;
class Gcd{

    static int gcd(int num1 , int num2){
        while(num2 != 0){
            int temp = num2;
            num2 = num1%num2;
            num1 = temp;
        }
        return num1;
    }


    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the first number : ");
        int num1 = sc.nextInt();
        System.out.println("Enter the second number : ");
        int num2 = sc.nextInt();

        System.out.println("The greatest common divisor is : " + gcd(num1, num2));
    }

}
```

```
C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java Gcd
Enter the first number :
45
Enter the second number :
54
The greatest common divisor is : 9
```

Time complexity: O(logn)
Space complexity: O(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
Ans.

```java
import java.util.*;
class Sqrt{

    static int findSquareRoot(int x){
        if(x<0){
            System.out.println("You entered a number for which sqaure root doesn't e
        }
        if(x==0||x==1){
            return x;
        }
        int result = x;
        while(result> x/result){
            result = (result+x/result)/2;
        }
        return result;
    }


    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("ENter the number : ");
        int x = sc.nextInt();

        int sqrt  = findSquareRoot(x);

        System.out.println("The sqaure root of " +x + " is " +sqrt );
    }
}
```

```
C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java Sqrt
ENter the number :
100
The sqaure root of 100 is 10
```

Time complexity: O(logn)
Space complexity: O(1)

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.

```java
import java.util.*;

class RepeatedCharacter{

    static void findRepeatedCharacter(String str){

    int[] charCount = new int[256];

    for(char c : str.toCharArray()){
        charCount[c]++;
    }

    System.out.print("Repeated characters: [");
    boolean first = true;
    for (char c = 0; c < charCount.length; c++) {
        if (charCount[c] > 1) {
            if (!first) {
                System.out.print(", ");
            }
                System.out.print("'" + c + "'");
                first = false;
            }
        }
    System.out.println("]");
}

    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the String : ");
        String text  = sc.nextLine();

        findRepeatedCharacter(text);
    }
}
```

```
C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java RepeatedCharacter
Enter the String :
abcdaefg
Repeated characters: ['a']
```

Time complexity: O(n)
Space complexity: O(1)

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.

```java
import java.util.*;

class FirstNRC{

    static Character findFirstNonRepeatedChar(String str){
        int[] charCount = new int[256];

    for(char c : str.toCharArray()){
        charCount[c]++;
    }

    for (char c : str.toCharArray()) {
        if (charCount[c] == 1) {
                return c;
            }
        }
        return ' ';

    }



    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the string : ");

        String txt = sc.nextLine();

        char res = findFirstNonRepeatedChar(txt);
        if(res != ' '){
            System.out.println("'"+res+"'");
        }
        else{
            System.out.println("No non-repeated character found.");
        }
    }
}
```

```
C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java FirstNRC
Enter the string :
hello
'h'
```

Time complexity: O(n)
Space complexity: O(1)

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.

```java
import java.util.Scanner;
class Palindrome{

    static boolean checkPalindrome(int num){
        if(num<0){
            return false;
        }
        int original = num;
        int reverse = 0;
        while(num>0){
            int lastDigit = num%10;
            reverse = reverse*10 + lastDigit;
            num = num/10;
        }
        return original == reverse;
    }


    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the integer for which you want ot check palindrome
        int num = sc.nextInt();

        boolean result = checkPalindrome(num);
        System.out.println(result);

    }
}
```

```
C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java Palindrome
Enter the integer for which you want ot check palindrome :
121
true

C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java Palindrome
Enter the integer for which you want ot check palindrome :
1342
false
```

Time complexity: O(logn)
Space complexity: O(1)

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.

```java
import java.util.*;

class LeapYear{

    static boolean isLeapYear(int year){
        return(year%4 == 0 && year%100 != 0)|| (year%400 == 0);
    }

    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the year : ");
        int year = sc.nextInt();

        boolean result = isLeapYear(year);
        System.out.println(result);
    }
}
```

```
C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java LeapYear
Enter the year :
2020
true

C:\Users\amang\OneDrive\Desktop\ADS\Assignments\Assignment 1>java LeapYear
Enter the year :
1990
false
```

Time complexity: O(1)
Space complexity: O(1)