Solution 01-12-23

1154. Day of the Year

```
// Online Java Compiler

// Use this editor to write, compile and run your Java code online


import java.util.Scanner;


public class DayOfYear {

   public static int dayOfYear(String date) {

      int[] daysInMonth = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

      String[] parts = date.split("-");

      int year = Integer.parseInt(parts[0]);

      int month = Integer.parseInt(parts[1]);

      int day = Integer.parseInt(parts[2]);


      int result = day;

      for (int i = 1; i < month; i++) {

         result += daysInMonth[i];

      }


      if (month > 2 && isLeapYear(year)) {

         result += 1;

      }


      return result;

   }


   public static boolean isLeapYear(int year) {

      return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

   }
```

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

        System.out.print("Enter a date (YYYY-MM-DD): ");
        String inputDate = scanner.nextLine();

        int result = dayOfYear(inputDate);
        System.out.println("Day of the year: " + result);
    }
}
```

Output:-

```
java -cp /tmp/5wiyLndx3U DayOfYear
Enter a date (YYYY-MM-DD): 2023-12-04
Day of the year: 338
```

965. Univalued Binary Tree

```
import java.util.*;

class TreeNode {
  int val;
  TreeNode left, right;

  TreeNode(int val) {
    this.val = val;
    this.left = this.right = null;
  }
}
```

```
public class UnivaluedBinaryTree {

 // Function to check if a binary tree is univalued
 public static boolean isUnivalTree(TreeNode root) {
  if (root == null) {
   return true;
  }

  return isUnivalTreeHelper(root, root.val);
 }

 private static boolean isUnivalTreeHelper(TreeNode node, int value) {
  if (node == null) {
   return true;
  }

  if (node.val != value) {
   return false;
  }

  return isUnivalTreeHelper(node.left, value) && isUnivalTreeHelper(node.right, value);
 }

 // Function to build a binary tree from user input
 public static TreeNode buildTree(Scanner scanner) {
  System.out.println("Enter the value for the node (enter -1 for null):");
  int value = scanner.nextInt();

  if (value == -1) {
   return null;
  }
```

```
    TreeNode root = new TreeNode(value);

    System.out.println("Enter left subtree for " + value + ":");
    root.left = buildTree(scanner);

    System.out.println("Enter right subtree for " + value + ":");
    root.right = buildTree(scanner);

    return root;
  }

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

    TreeNode root = buildTree(scanner);

    boolean result = isUnivalTree(root);

    if (result) {
      System.out.println("The binary tree is univalued.");
    } else {
      System.out.println("The binary tree is not univalued.");
    }
  }
}
```

Outpu

```
PS C:\Users\Ajeet\Desktop\java> javac UnivaluedBinaryTree.java
PS C:\Users\Ajeet\Desktop\java> java UnivaluedBinaryTree
Enter the value for the node (enter -1 for null):
```

```
1
Enter left subtree for 1:
Enter the value for the node (enter -1 for null):
2
Enter left subtree for 2:
Enter the value for the node (enter -1 for null):
-1
Enter right subtree for 2:
Enter the value for the node (enter -1 for null):
-1
Enter right subtree for 1:
Enter the value for the node (enter -1 for null):
3
Enter left subtree for 3:
Enter the value for the node (enter -1 for null):
-1
Enter right subtree for 3:
Enter the value for the node (enter -1 for null):
-1
The binary tree is not univalued.
```