Solution 03-11-23

326. Power of Three

```
import java.util.Scanner;

public class PowerOfThree {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        boolean isPowerOfThree = isPowerOfThree(number);

        if (isPowerOfThree) {
            System.out.println(number + " is a power of three.");
        } else {
            System.out.println(number + " is not a power of three.");
        }
    }

    public static boolean isPowerOfThree(int n) {
        if (n <= 0) {
            return false;
        }

        while (n > 1) {
            if (n % 3 != 0) {
                return false;
            }
            n /= 3;
        }
```

```
        return true;
    }
}
```

output

```
java -cp /tmp/9xMiWSWqb4 PowerOfThree
Enter a number: 9
9 is a power of three.
```

306. Additive Number

```
import java.util.Scanner;

public class AdditiveNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string of digits: ");
        String input = scanner.nextLine();

        boolean isAdditive = isAdditiveNumber(input);

        if (isAdditive) {
            System.out.println(input + " true");
        } else {
            System.out.println(input + " false");
        }
    }

    public static boolean isAdditiveNumber(String num) {
```

```
        int n = num.length();
        for (int i = 1; i < n; i++) {
            for (int j = i + 1; j < n; j++) {
                String num1 = num.substring(0, i);
                String num2 = num.substring(i, j);

                if ((num1.length() > 1 && num1.charAt(0) == '0') || (num2.length() > 1 && num2.charAt(0)
== '0')) {
                    continue; // Numbers with leading zeros are not valid
                }

                if (isAdditiveSequence(num, num1, num2, j)) {
                    return true;
                }
            }
        }

        return false;
    }

    public static boolean isAdditiveSequence(String num, String num1, String num2, int startIndex) {
        if (startIndex == num.length()) {
            return true; // The entire string has been processed, and it's an additive sequence
        }

        String sum = addStrings(num1, num2);
        if (!num.startsWith(sum, startIndex)) {
            return false; // The next part of the string doesn't match the sum of num1 and num2
        }

        // Recursively check the rest of the string
```

```
        return isAdditiveSequence(num, num2, sum, startIndex + sum.length());
    }

    public static String addStrings(String num1, String num2) {
        int carry = 0;
        StringBuilder result = new StringBuilder();
        int i = num1.length() - 1;
        int j = num2.length() - 1;

        while (i >= 0 || j >= 0 || carry > 0) {
            int digit1 = i >= 0 ? num1.charAt(i--) - '0' : 0;
            int digit2 = j >= 0 ? num2.charAt(j--) - '0' : 0;

            int sum = digit1 + digit2 + carry;
            carry = sum / 10;
            result.insert(0, sum % 10);
        }

        return result.toString();
    }
}
```

Output

```
java -cp /tmp/9xMiWSWqb4 AdditiveNumber
Enter a string of digits: 112358
112358 true
```