Leetcode

Leetcode Solution 1 :-

```
class Solution {
    public String fractionAddition(String expression) {
        int x = 0, y = 6 * 7 * 8 * 9 * 10;
        if (Character.isDigit(expression.charAt(0))) {
            expression = "+" + expression;
        }
        int i = 0, n = expression.length();
        while (i < n) {
            int sign = expression.charAt(i) == '-' ? -1 : 1;
            ++i;
            int j = i;
            while (j < n && expression.charAt(j) != '+' &&
expression.charAt(j) != '-') {
                ++j;
            }
            String s = expression.substring(i, j);
            String[] t = s.split("/");
            int a = Integer.parseInt(t[0]), b = Integer.parseInt(t[1]);
```

```
            x += sign * a * y / b;
            i = j;
        }
        int z = gcd(Math.abs(x), y);
        x /= z;
        y /= z;
        return x + "/" + y;
    }

    private int gcd(int a, int b) {
        return b == 0 ? a : gcd(b, a % b);
    }
}
```

Leetcode Solution 2 :-

```
class Solution {
    public int findLHS(int[] nums) {
        Map<Integer, Integer> counter = new HashMap<>();
        for (int num : nums) {
            counter.put(num, counter.getOrDefault(num, 0) + 1);
        }
        int ans = 0;
        for (int num : nums) {
            if (counter.containsKey(num + 1)) {
                ans = Math.max(ans, counter.get(num) + counter.get(num +
1));
            }
        }
        return ans;
    }
}
```