[Roman to Integer](https://leetcode.com/problems/roman-to-integer/)

int getValue(char c) {

    switch (c) {

        case 'I': return 1;

        case 'V': return 5;

        case 'X': return 10;

        case 'L': return 50;

        case 'C': return 100;

        case 'D': return 500;

        case 'M': return 1000;

        default: return -1;

    }

}

int romanToInt(char \* s){

int result = 0;

    int prev = -1;

    for (int i = strlen(s) - 1; i >= 0; i--) {

        int curr = getValue(s[i]);

        if (curr ==-1) {

            return -1;

        }

        if (prev == -1 || prev <= curr) {

            result += curr;

        } else {

            result -= curr;

        }

        prev = curr;

    }

    return result;

}

[Longest Common Prefix](https://leetcode.com/problems/longest-common-prefix/)

class Solution:

    def longestCommonPrefix(self, strs: List[str]) -> str:

        if not strs:

             return ""

        if len(strs) == 1:

             return strs[0]

        strs.sort()

        first = strs[0]

        last = strs[-1]

        for i, char in enumerate(first):

             if char != last[i]:

                  return first[:i]

        return first

[Valid Palindrome](https://leetcode.com/problems/valid-palindrome/)

import re

class Solution:

    def isPalindrome(self, s: str) -> bool:

        s = re.sub(r'[^a-zA-Z0-9]', '', s)

        s = s.lower()

        return s == s[::-1]

[Valid Anagram](https://leetcode.com/problems/valid-anagram/)

class Solution {

    public boolean isAnagram(String s, String t) {

        int[] arr=new int[26];

        for(int i=0;i<s.length();i++)

            arr[s.charAt(i)-'a']++;

        for(int i=0;i<t.length();i++)

            arr[t.charAt(i)-'a']--;

        for(int i=0;i<26;i++)

            if(arr[i]!=0)

                return false;

        return true;

    }

}

[Reverse Words in a String](https://leetcode.com/problems/reverse-words-in-a-string/)

class Solution:

    def reverseWords(self, s: str) -> str:

        return " ".join(s.split()[::-1])