

QUESTION BANK DOCKET

**Table of Contents**

[1 Source 2](#_Toc253285289)

[2 Category 3](#_Toc965934774)

[3 Tech Area 3](#_Toc613746458)

[4 Complexity 3](#_Toc904143571)

[5 Question 3](#_Toc1426514880)

[6 Answer 3](#_Toc441207452)

[7 References 3](#_Toc1838997797)

# Source

Leet Code

# Category

|  |
| --- |
| DOT NET |

# Tech Area

C#

# Complexity

Simple

# Question

1. Given an integer x, return true *if* x *is a* ***palindrome****, and* false *otherwise*.
2. Given an input string s and a pattern p, implement regular expression matching with support for '.' and '\*' where:

* '.' Matches any single character.
* '\*' Matches zero or more of the preceding element. The matching should cover the **entire** input string (not partial).

1. Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0. Notice that the solution set must not contain duplicate triplets.
2. Given an integer array nums and an integer val, remove all occurrences of val in nums **in-place**. The relative order of the elements may be changed. Since it is impossible to change the length of the array in some languages, you must instead have the result be placed in the **first part** of the array nums. More formally, if there are k elements after removing the duplicates, then the first k elements of nums should hold the final result. It does not matter what you leave beyond the first k elements. Return k *after placing the final result in the first* k *slots of* nums. Do **not** allocate extra space for another array. You must do this by **modifying the input array in-place** with O(1) extra memory.
3. Given two strings needle and haystack, return the index of the first occurrence of needle in haystack, or -1 if needle is not part of haystack.
4. You are given the heads of two sorted linked lists list1 and list2.Merge the two lists in a one **sorted** list. The list should be made by splicing together the nodes of the first two lists. Return *the head of the merged linked list*.
5. Given n pairs of parentheses, write a function to *generate all combinations of well-formed parentheses*.
6. You are given an array of k linked-lists lists, each linked-list is sorted in ascending order. *Merge all the linked-lists into one sorted linked-list and return it.*
7. Given a string containing just the characters '(' and ')', return *the length of the longest valid (well-formed) parentheses* Substring
8. Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order. You must write an algorithm with O (log n) runtime complexity.

# Answer

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

# References

[Discussion Boards, Comments, Links, etc..]x

https://leetcode.com/problemset/all/