Fresher DSA questions

1. Two Sum:
   Given an array of integers `nums` and an integer `target`, return the indices of the two numbers such that they add up to the `target`.
   Example:

```
Input: nums = [2, 7, 11, 15], target = 9
Output: [0, 1]  // Explanation: nums[0] + nums[1] = 2 + 7 = 9
```

2. Reverse Integer:
   Given a 32-bit signed integer `x`, reverse the digits of the integer.
   Example:

```
Input: x = 123
Output: 321
```

3. Valid Parentheses:
   Given a string `s` containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.
   Example:

```
Input: s = "{[()]}[()]{[]}"
Output: true
```

4. Merge Two Sorted Lists:
   Merge two sorted linked lists and return it as a new sorted list.
   Example:

```
Input: l1 = 1->2->4, l2 = 1->3->4
Output: 1->1->2->3->4->4
```

5. Maximum Subarray:
   Given an integer array `nums`, find the contiguous subarray (containing at least one

number) that has the largest sum and return its sum.
Example:

```
Input: nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4]
Output: 6  // Explanation: [4, -1, 2, 1] has the largest sum = 6.
```

6. Palindrome Number:
   Determine whether an integer is a palindrome.
   Example:

```
Input: x = 121
Output: true
```

7. Climbing Stairs:
   You are climbing a staircase. It takes `n` steps to reach the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?
   Example:

```
Input: n = 3
Output: 3  // Explanation: There are three ways to climb to the top: 1->1->1, 1->2, a
nd 2->1.
```

8. Longest Common Prefix:
   Write a function to find the longest common prefix string amongst an array of strings.
   Example:

```
Input: strs = ["flower", "flow", "flight"]
Output: "fl"
```

9. Valid Anagram:
   Given two strings `s` and `t`, return true if `t` is an anagram of `s`, and false otherwise.
   Example:

```
Input: s = "anagram", t = "nagaram"
Output: true
```

10. Binary Tree Level Order Traversal:
    Given the root of a binary tree, return the level order traversal of its nodes' values.
    Example:

```
Input: root = [3, 9, 20, null, null, 15, 7]
        3
       / \\
      9  20
        /  \\
       15   7
Output: [[3], [9, 20], [15, 7]]
```