Here’s a curated list of **20 Python coding questions** involving lists, ranging from basic to advanced levels. These questions cover essential operations, problem-solving, and algorithmic concepts using Python lists:

**Basic Python List Questions**

1. **Reverse a List**
   * Write a function to reverse a given list without using the reverse() method.
2. **Find the Largest Element in a List**
   * Write a program to find the largest number in a list.
3. **Find the Second Largest Element**
   * Extend the above program to find the second largest number.
4. **Check if a List is Sorted**
   * Write a function to check whether a list is sorted in ascending order.
5. **Remove Duplicates**
   * Write a program to remove duplicates from a list while preserving the order.
6. **Flatten a Nested List**
   * Write a program to flatten a nested list (e.g., [[1, 2], [3, 4]] → [1, 2, 3, 4]).
7. **Rotate a List**
   * Rotate a list by k positions to the right (e.g., [1, 2, 3, 4, 5] rotated by 2 becomes [4, 5, 1, 2, 3]).
8. **Count the Frequency of Elements**
   * Write a program to count the frequency of each element in a list.
9. **Split a List in Half**
   * Divide a list into two approximately equal halves.
10. **Concatenate Two Lists**
    * Write a program to concatenate two lists without using the + operator.

**Intermediate Python List Questions**

1. **Find the Intersection of Two Lists**
   * Write a function to find the common elements between two lists.
2. **Find the Union of Two Lists**
   * Write a function to compute the union of two lists without duplicates.
3. **Find Pairs with a Given Sum**
   * Given a list of numbers and a target sum, find all pairs of numbers in the list that add up to the target sum.
4. **Sort a List of Tuples**
   * Sort a list of tuples based on the second element in each tuple (e.g., [(1, 3), (4, 1), (2, 2)] → [(4, 1), (2, 2), (1, 3)]).
5. **Find the Missing Number**
   * Given a list of integers from 1 to n with one number missing, find the missing number.
6. **Find Continuous Subarray with a Given Sum**
   * Write a program to find a continuous subarray in a list that sums up to a given number.
7. **Generate All Permutations of a List**
   * Write a function to generate all permutations of a given list.
8. **Find the Majority Element**
   * Given a list of integers, find the element that appears more than n/2 times (if such an element exists).
9. **Find the Longest Increasing Subsequence**
   * Write a program to find the longest increasing subsequence in a list.
10. **Group Anagrams**
    * Given a list of strings, group them into lists of anagrams (e.g., ["eat", "tea", "tan", "ate", "nat", "bat"] → [["eat", "tea", "ate"], ["tan", "nat"], ["bat"]]).

**Hard Python List Questions**

1. **Subarray with Maximum Sum (Kadane's Algorithm)**
   * Given a list of integers, find the contiguous subarray with the largest sum. (Example: [-2, 1, -3, 4, -1, 2, 1, -5, 4] → 6, from [4, -1, 2, 1]).
2. **Trapping Rainwater**
   * Given a list representing the heights of bars, calculate the amount of water that can be trapped between them. (Example: [0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1] → 6).
3. **Longest Consecutive Sequence**
   * Given an unsorted list of integers, find the length of the longest consecutive elements sequence. (Example: [100, 4, 200, 1, 3, 2] → 4, sequence: [1, 2, 3, 4]).
4. **3Sum Problem**
   * Find all unique triplets in the list that add up to zero. (Example: [-1, 0, 1, 2, -1, -4] → [[-1, -1, 2], [-1, 0, 1]]).
5. **Sort Colors (Dutch National Flag Problem)**
   * Given a list with values 0, 1, and 2 (representing colors), sort the list in-place. (Example: [2, 0, 2, 1, 1, 0] → [0, 0, 1, 1, 2, 2]).
6. **Maximum Product Subarray**
   * Find the contiguous subarray within a list of integers that has the largest product. (Example: [2, 3, -2, 4] → 6, from [2, 3]).
7. **Sliding Window Maximum**
   * Given a list and a window size k, return the maximum value in each sliding window of size k. (Example: [1,3,-1,-3,5,3,6,7], k=3 → [3, 3, 5, 5, 6, 7]).
8. **Find Median of Two Sorted Lists**
   * Given two sorted lists, find the median of the combined sorted list in O(log(min(m, n))) time complexity.
9. **Merge Overlapping Intervals**
   * Given a list of intervals, merge overlapping ones. (Example: [[1,3],[2,6],[8,10],[15,18]] → [[1,6],[8,10],[15,18]]).
10. **Kth Largest Element in an Unsorted List**
    * Write a program to find the kth largest element in an unsorted list using an optimal approach (e.g., quickselect).