*Python Questions Practical*

Note: In every program you have to use class, function and proper Exception handling.

**1. Take the input from user and find whether number is prime or not?**

Prime numbers are number which are divide by 1 or themselves only.

Example: 13 is a prime number.

Example: 97 is a prime number

**2. Reverse the string using for loop.**

string = "Python"

Expected output = "nohtyP"

**3. Palindrome Check (String Problem)**

Write a function to check whether a given string is a palindrome (a string that reads the same forward and backward is called palindrome).

String = “mom” # if we reverse this string, we get the same Output “mom”

Expected output: string is a palindrome.

**4. Calculate factorial of a number**

Write a function calculate\_factorial that takes a number as input and returns its factorial. Handle cases where the input is not a non-negative integer or zero.

Example: If number is 5 factorials will be (5x4x2x3x2x1=120)

Expected output: 120 if number is 5

**5. Find the Second Largest Element in a List using for loop.**

Write a function to find the second largest element in a list without using built-in sorting.

Example  
Input: [7, 5, 8, 2, 10, 9]  
Expected Output: 9

Example  
Input: [4, 4, 4, 4]  
Expected Output: None

**6. Check for Anagram**

* Write a function to check if two input strings are anagrams of each other.
* An anagram is a word or phrase that is formed by rearranging the letters of another word or phrase. To be anagrams, two strings must contain the exact same characters in the same quantities, but in any order.
* Example: "listen" and "silent" are anagrams.

**7. Count Vowels in a String**

* Take a string as input and count the number of vowels (a, e, i, o, u) in it.
* Ignore case (i.e., count both uppercase and lowercase vowels).
* Example: “Python” word has 1 vowel sound(o)

**8. Sum of Digits**

* Write a function that takes an integer as input and returns the sum of its digits.
* Example: sum\_of\_digits(123) -> 6

**9. Fibonacci Sequence (Iterative)**

* Write a function that prints the first n numbers in the Fibonacci sequence.
* Use an iterative approach.
* Example  
  Input: n = 6  
  Expected Output: 0, 1, 1, 2, 3, 5
* Example  
  Input: n = 1  
  Expected Output: 0

**10. Remove Duplicates from a List**

* Write a function that removes duplicates from a list while maintaining the original order.
* Example  
  Input: [1, 3, 2, 3, 4, 1, 5]  
  Expected Output: [1, 3, 2, 4, 5]
* Example  
  Input: [4, 4, 4, 4]  
  Expected Output: [4]

**11. Check for Leap Year**

* Write a function to check if a given year is a leap year.
* A leap year is divisible by 4, but not by 100, unless also divisible by 400.
* Divisible by 4: If a year can be evenly divided by 4 (like 2020), it might be a leap year.
* Not Divisible by 100: However, if that year can also be evenly divided by 100 (like 1900), then it's **not** a leap year.
* Unless Divisible by 400: If the year is divisible by 100, it could still be a leap year if it can also be evenly divided by 400 (like 2000).
* Example  
  Input: year = 2020  
  Expected Output: True (2020 is a leap year)
* Example  
  Input: year = 1900  
  Expected Output: False (1900 is not a leap year)
* Example  
  Input: year = 2000  
  Expected Output: True (2000 is a leap year)

**12. Calculate Power without Using \*\* Operator**

* Write a function that calculates xy (x raised to the power of y) without using Python’s power operator (\*\*).
* Example  
  Input: x = 2, y = 3  
  Expected Output: 8 (since 23 =8)

**13. Generate List of Even Numbers**

* Create a function that generates a list of even numbers up to a given number.
* Example  
  Input: n = 10  
  Expected Output: [2, 4, 6, 8, 10]
* Example  
  Input: n = 7  
  Expected Output: [2, 4, 6]

**14. Binary Search Implementation**

* Write a function that performs binary search on a sorted list and returns the index of the target element or -1 if not found.
* Example  
  Input: sorted\_list = [1, 2, 3, 4, 5, 6], target = 4  
  Expected Output: 3 (since 4 is at index 3)
* Example  
  Input: sorted\_list = [10, 20, 30, 40, 50], target = 25  
  Expected Output: -1 (since 25 is not in the list)

**15. Reverse Words in a Sentence**

* Write a function to reverse the order of words in a sentence, keeping the words themselves intact (sentence should not be changed or altered in any way; they should remain exactly as they are,).

Example  
Input: "Python is fun"  
Expected Output: "fun is Python"

**16. Calculate GCD Using Euclidean Algorithm**

* Write a function to find the Greatest Common Divisor (GCD) of two numbers using recursion.
* Example  
  Input: a = 48, b = 18  
  Expected Output: 6 (the GCD of 48 and 18 is 6)
* Example  
  Input: a = 56, b = 98  
  Expected Output: 14 (the GCD of 56 and 98 is 14)

**17. Convert Decimal to Binary**

* Write a function to convert a given decimal number to its binary representation.
* Example  
  Input: decimal = 10  
  Expected Output: 1010 (the binary representation of 10 is 1010)
* Example  
  Input: decimal = 5  
  Expected Output: 101 (the binary representation of 5 is 101)

**18. Factorial Using Recursion**

* Write a recursive function to calculate the factorial of a given number.
* Example  
  Input: n = 5  
  Expected Output: 120 (since 5×4×3×2×1=120)
* Example  
  Input: n = 0  
  Expected Output: 1

**19. Find Largest Element in a List Using for Loop**

* Write a function to find the largest element in a list without using built-in max functions.
* Example  
  Input: [3, 5, 7, 2, 8, 1]  
  Expected Output: 8 (the largest element in the list is 8)
* Example  
  Input: [-1, -5, -3, -4]  
  Expected Output: -1 (the largest element in the list is -1)

**20. Count Occurrences of Each Character in a String**

* Take a string as input and count the occurrences of each character.
* Example:  
  **Input**: "hello"  
  **Expected Output**: { 'h': 1, 'e': 1, 'l': 2, 'o': 1 }
* Example:  
  **Input**: "apple"  
  **Expected Output**: { 'a': 1, 'p': 2, 'l': 1, 'e': 1 }

**21. Find Intersection of Two Lists**

* Write a function that returns the intersection of two lists without using set operations.
* Example:  
  Input: list1 = [1, 2, 3, 4, 5], list2 = [4, 5, 6, 7, 8]  
  Expected Output: [4, 5]
* Example:  
  Input: list1 = ['apple', 'banana', 'cherry'], list2 = ['cherry', 'date', 'apple']  
  Expected Output: ['apple', 'cherry']

**22. Remove All Whitespace from a String**

* Write a function that removes all whitespace characters (spaces, tabs, newlines) from a given string.
* Example:  
  **Input**: "Hello, World!"  
  **Expected Output**: "Hello,World!"
* Example:  
  **Input**: " Python is fun "  
  **Expected Output**: "Pythonisfun"

**23. Calculate Sum of a List of Numbers Using Recursion**

* Write a recursive function to calculate the sum of a list of numbers.
* Example:  
  **Input**: [1, 2, 3, 4, 5]  
  **Expected Output**: 15
* Example:  
  **Input**: [10, 20, 30]  
  **Expected Output**: 60

**24. Find All Prime Numbers up to n**

* Write a function that returns a list of all prime numbers up to a given number n.
* Example:  
  **Input**: n = 10  
  **Expected Output**: [2, 3, 5, 7]
* Example:  
  **Input**: n = 20  
  **Expected Output**: [2, 3, 5, 7, 11, 13, 17, 19]

**25. Merge Two Sorted Lists**

* Write a function to merge two sorted lists into a single sorted list.
* Example:  
  **Input**: list1 = [1, 3, 5], list2 = [2, 4, 6]  
  **Expected Output**: [1, 2, 3, 4, 5, 6]
* Example:  
  **Input**: list1 = [10, 20, 30], list2 = [15, 25, 35]  
  **Expected Output**: [10, 15, 20, 25, 30, 35]

**26. Check if List is Sorted**

* Write a function that checks if a list is sorted in ascending order.
* Example:  
  **Input**: [1, 2, 3, 4, 5]  
  **Expected Output**: True
* Example:  
  **Input**: [5, 3, 4, 2, 1]  
  **Expected Output**: False

**27. Count Frequency of Each Word in a Text File**

* Write a program to read a text file and count the frequency of each word.

Example: File content

Hello world

Hello again

Goodbye world

Expected Output:

Hello: 2

world: 2

again: 1

Goodbye: 1

28. Find Missing Number in Consecutive List

* Write a function to find the missing number in a list of consecutive numbers.
* Example:  
  **Input**: [1, 2, 3, 5, 6]  
  **Expected Output**: 4
* Example:  
  **Input**: [10, 11, 12, 14, 15]  
  **Expected Output**: 13

**29. Calculate Sum of Squares of First n Natural Numbers**

* Write a function to calculate the sum of squares of the first n natural numbers.
* Example:  
  **Input**: n = 3  
  **Expected Output**: 14  
  *(Calculation:* 12 + 22 + 32 =14*)*

**30. Reverse Integer**

* Write a function that takes an integer and returns its digits reversed.
* Example:  
  **Input**: 12345  
  **Expected Output**: 54321
* Example:  
  **Input**: -6789  
  **Expected Output**: -9876

**31. Implement a Stack using List**

* Create a stack with push, pop, and peek operations using a list.

Stack: A stack is a linear data structure that follows the Last In, First Out (LIFO) principle. This means that the last element added to the stack is the first one to be removed. Think of it like a stack of plates; you can only add or remove plates from the top.

List: A list is a more general data structure that can store a collection of items. It allows for random access and can be used to store elements in any order. Lists do not have strict rules about how elements are added or removed.

* Example:  
  Stack Operations:
* Push: stack.push(10)
* Push: stack.push(20)
* Push: stack.push(30)
* Peek: stack.peek()  
  Expected Output: 30
* Pop: stack.pop()  
  Expected Output: 30
* Peek: stack.peek()  
  Expected Output: 20
* Pop: stack.pop()  
  Expected Output: 20
* Pop: stack.pop()  
  Expected Output: 10
* Pop: stack.pop()  
  Expected Output: None (or an indication that the stack is empty)

**32. Find the Longest Word in a Sentence**

* Write a function that finds and returns the longest word in a given sentence.

Example:  
**Input**: "Python programming is fun and interesting"  
Expected Output: "programming"

**33. Check if a Number is a Power of 2**

* Write a function that checks if a given number is a power of 2.
* Example:  
  **Input**: 16  
  **Expected Output**: True  
  *(Explanation:* 24=16*)*
* Example:  
  **Input**: 18  
  **Expected Output**: False  
  *(Explanation: 18 is not a power of 2)*

**34. Flatten a Nested List**

* Write a function to flatten a list that contains nested lists of integers.
* Example:  
  **Input**: [1, [2, [3, 4], 5], 6]  
  **Expected Output**: [1, 2, 3, 4, 5, 6]
* Example:  
  **Input**: [[1, 2], [3, [4, 5]], 6]  
  **Expected Output**: [1, 2, 3, 4, 5, 6]

**35. Find Pairs with Given Sum in a List**

* Write a function to find all pairs in a list that sum up to a given number.
* **Example**
* **Input:**

List: [1, 2, 3, 4, 5, 6]

Target Sum: 7

* **Expected Output:**

[(1, 6), (2, 5), (3, 4)]

In this example, the pairs (1, 6), (2, 5), and (3, 4) all add up to the target sum of 7.

**36. Check if Two Strings are Rotations of Each Other**

* Write a function to check if one string is a rotation of another.
* Example: “abcd” and “dabc” are rotations.

To understand why, here’s how rotation works in this context:

If you rotate "abcd" by shifting its characters cyclically (moving some characters from the start to the end), you get:

1. Rotate by 1 position: "bcda"(remove first char and put it at the end)
2. Rotate by 2 positions: "cdab" (remove first 2 char and put it at the end)
3. Rotate by 3 positions: "dabc" (remove first 3 char and put it at end)

**37. Implement Queue using List**

* Create a queue with enqueue, dequeue, and peek operations using a list.

Example Scenario

1. Enqueue Operations:
   * Add elements to the queue in the order: 10, 20, 30.
2. Dequeue Operation:
   * Remove the front element of the queue (FIFO principle).
3. Peek Operation:
   * View the element at the front of the queue without removing it.

**Expected Output**

Given the above operations, the sequence would look like this:

* After enqueuing 10, 20, and 30:  
  Queue: [10, 20, 30]
* After calling peek (without removing any item):  
  Output: 10
* After calling dequeue twice:  
  Queue: [30]
* After calling peek again:  
  Output: 30
* After calling dequeue one more time (emptying the queue):  
  Queue: []
* If attempting to dequeue on an empty queue:  
  Output: "Queue is empty, cannot dequeue."

**38. Convert List of Tuples to Dictionary**

* Write a function to convert a list of tuples into a dictionary.

Example:

* Input:  
  [(‘a’, 1), (‘b’, 2), (‘c’, 3), (‘d’, 4)]
* Expected Output:  
  {'a': 1, 'b': 2, 'c': 3, 'd': 4}

**39. Find the Median of a List of Numbers**

* Write a function to calculate the median of a list of numbers.

Example:

* Input:  
  [1, 3, 5, 7, 9]

**Expected Output:**  
5

* Input:  
  [2, 4, 6, 8]

Expected Output:  
5.0

**40. Sort Dictionary by Value**

* Write a function to sort a dictionary by its values.

Example:

* Input:  
  {'apple': 5, 'banana': 2, 'cherry': 8, 'date': 3}
* Expected Output (sorted by values in ascending order):  
  {'banana': 2, 'date': 3, 'apple': 5, 'cherry': 8}

**41. Generate a List of Prime Numbers Using Sieve of Eratosthenes**

* Write a function to generate all prime numbers up to a given number using the Sieve of Eratosthenes.

Example:

* Input:  
  n = 20

Expected Output:  
[2, 3, 5, 7, 11, 13, 17, 19]

**42. Convert Binary to Decimal**

* Write a function to convert a binary number (as a string) to a decimal.

Example:

* Input:  
  "1011"

Expected Output:  
11

**43. Check if Subsequence Exists in List**

* Write a function to check if a smaller list is a subsequence of a larger list.

Example:

* Input:  
  larger\_list = [1, 3, 5, 7, 9]  
  smaller\_list = [3, 7]

Expected Output:  
True

* Input:  
  larger\_list = [1, 3, 5, 7, 9]  
  smaller\_list = [3, 8]

Expected Output:  
False

**44. Find Common Elements in Three Lists**

* Write a function to find elements common in three lists.

Example:

* Input:  
  list1 = [1, 2, 3, 4, 5]  
  list2 = [3, 4, 5, 6, 7]  
  list3 = [5, 6, 7, 8, 9]
* Expected Output:  
  [5]

**45. Swap Two Variables Without Temporary Variable**

* Write a program to swap the values of two variables without using a temporary variable.

Example:

* Input:  
  a = 5  
  b = 10
* Expected Output (after swapping):  
  a should become 10  
  b should become 5

**46. Remove Nth Occurrence of a Character from a String**

* Write a function that removes the nth occurrence of a character from a string.

Example:

* Input**:**  
  string = "example example example"  
  char = 'e'  
  n = 2
* Expected Output**:**  
  "example xample example"

**47. Find the First Non-Repeated Character in a String**

* Write a function to return the first non-repeated character in a string.

**Example:**

* Input:  
  string = "swiss"

Expected Output**:**  
'w'

* Input:  
  string = "repeated"

Expected Output:  
'r'

Explanation:  
The function should scan the input string and return the first character that does not repeat. If all characters are repeated, it should return None or an appropriate message indicating that there are no non-repeated characters.

**48. Check if String Contains Only Digits**

* Write a function that checks if a string contains only numeric characters.

**Example:**

* Input:  
  string = "123456"
* Expected Output:  
  True
* Input:  
  string = "123a56"
* Expected Output:  
  False

**49. Check if Number is Armstrong Number**

* Write a function that checks if a number is an Armstrong number.
* Example: 153 is an Armstrong number

because 13+53+33=153.

Explanation**:**  
An Armstrong number (also known as a narcissistic number) for a given number of digits is an integer such that the sum of its own digits raised to the power of the number of digits is equal to the number itself.

**50. Generate All Subsets of a Set**

* Write a function to generate all possible subsets of a given set (list) of numbers.

**Example:**

* Input**:**  
  numbers = [1, 2, 3]
* Expected Output**:**  
  [[], [1], [2], [3], [1, 2], [1, 3], [2, 3], [1, 2, 3]]