Practical:

SUBJECT NAME: Python Programming Lab **Credit: 2**
SUBJECT CODE: BCAC391

List of Practical:

1. Fizz Buzz: Write a program that prints the numbers from 1 to 100. But for multiples of three, print "Fizz" instead of the number, and for the multiples of five, print "Buzz". For numbers that are multiples of both three and five, print "Fizz Buzz".
2. Palindrome Checker: Write a function to determine if a given string is a palindrome (reads the same forwards and backwards). Ignore spaces, punctuation, and capitalization.
3. Factorial Calculation: Write a function to calculate the factorial of a given number recursively.
4. Prime Number Generator: Write a function to generate a list of prime numbers up to a given number using the Sieve of Eratosthenes algorithm.
5. Word Count: Write a program that takes a string as input and counts the frequency of each word in the string. Ignore case and punctuation.
6. Reverse a Linked List: Implement a function to reverse a singly linked list in-place.
7. Binary Search: Implement the binary search algorithm to find the index of a given element in a sorted list.
8. Anagram Checker: Write a function to determine if two strings are anagrams of each other (contain the same characters in a different order).
9. Matrix Transpose: Write a function to transpose a given matrix (convert rows to columns and vice versa).
10. Tower of Hanoi: Implement the Tower of Hanoi puzzle using recursion.

List:

1. **Sum of List Elements**: Write a program that calculates the sum of all elements in a list of numbers.
2. **Maximum and Minimum Element in List**: Write a program to find the maximum and minimum elements in a list.
3. **List Reversal**: Write a program to reverse a given list.
4. **List Sorting**: Write a program to sort a list of numbers in ascending or descending order.
5. **List Filtering**: Write a program to filter out even or odd numbers from a list.
6. **List Concatenation**: Write a program to concatenate two lists into one.
7. **List Intersection**: Write a program to find the intersection of two lists (i.e., elements that appear in both lists).
8. **List Union**: Write a program to find the union of two lists (i.e., all unique elements from both lists).
9. **List Flattening**: Write a program to flatten a nested list (i.e., convert a list of lists into a single list).
10. **List Element Removal**: Write a program to remove all occurrences of a specific element from a list.

11. **List Rotation**: Write a program to rotate a list by a given number of positions.
12. **List Comprehensions**: Write a program to generate a new list based on a given list using list comprehensions (e.g., square each element of a list).

Dictionary:

Word Frequency Counter:
Write a program that takes a string as input and counts the frequency of each word using a dictionary. Ignore case and punctuation.

Merge Two Dictionaries:
Write a function to merge two dictionaries into one, where the values of duplicate keys are added together.

Dictionary Key Sort:
Write a function to sort the keys of a dictionary in alphabetical order and return a new dictionary with the sorted keys.

Nested Dictionary Access:
Write a function to access a value in a nested dictionary given a list of keys. For example, given the dictionary {'a': {'b': {'c': 1}}} and the keys ['a', 'b', 'c'], the function should return 1.

Dictionary Inversion:
Write a function to invert a dictionary, where the keys become values and the values become keys. Assume that the values are unique.

Dictionary Intersection:
Write a function to find the intersection of two dictionaries (i.e., keys that appear in both dictionaries) and return a new dictionary with the common keys and their values.

SUGGESTED READING:

1. Python **Programming: A Modular Approach" by Sheetal Taneja and Naveen Kumar** (Publisher: Oxford University Press India)
2. **Python for Beginners: A Step-by-Step Guide to Learn Python from Zero with Hands-on Exercises" by Ajit Kumar** (Publisher: BPB Publications)
3. "**Python: A Practical Introduction to Programming" by Subin Siby** (Publisher: BPB Publications)
4. "**Python Programming: Problems and Solutions" by S.S. Srivastava and M.H. Khan** (Publisher: Khanna Publishers)
5. "**Python Programming: A Beginner's Guide to Learn Python in 7 Days" by Darshan Patel** (Publisher: BPB Publications)