Date	Sep-19-2024	Session No	5

Topic : Programming / Problem solving

Assignment Questions

Level 1

- Write a function that takes a list as input and returns the sum of all elements in the list.
- Write a function that takes a list of numbers and returns the largest number in the list.
- Write a function that takes a list and returns the reverse of the list.
- Write a function that removes all duplicates from a list.
- Write a function that takes two lists and returns their intersection (common elements).

Level 2

- Write a function that takes a list of strings and returns a list of the strings sorted by length.
- Write a function that takes a list and a number n and returns the first n elements of the list.
- Write a function that takes a list of numbers and returns a new list containing only the even numbers.
- Write a function that flattens a nested list of lists.
- Write a function that takes a list of numbers and returns the second largest number.

Level 3

- Write a function that takes a list and returns all possible subsets of that list.
- Write a function that takes a list and returns the list rotated to the right by n places.

- Write a function that takes a list of numbers and returns `True` if the list is sorted in ascending order, otherwise `False`.
- Write a function that returns the most frequent element in a list. If there are ties, return any one of the most frequent elements.
- Write a function that takes two lists and returns `True` if they are permutations of each other.

Solutions

Level 1:

Write a function that takes a list as input and returns the sum of all elements in the list.

```
def sum_list(lst):
    return sum(lst)
```

Write a function that takes a list of numbers and returns the largest number in the list.

```
def max_in_list(lst):
    return max(lst)
```

Write a function that takes a list and returns the reverse of the list.

```
def reverse_list(lst):
    return lst[::-1]
```

Write a function that removes all duplicates from a list.

```
def remove_duplicates(lst):
    return list(set(lst))
```

Write a function that takes two lists and returns their intersection (common elements).

```
 def list_intersection(lst1, lst2):
    return list(set(lst1) & set(lst2))
```

Level 2:

Write a function that takes a list of strings and returns a list of the strings sorted by length.

```
def sort_by_length(lst):
    return sorted(lst, key=len)
```

Write a function that takes a list and a number n and returns the first n elements of the list.

```
def first_n_elements(lst, n):
    return lst[:n]
```

Write a function that takes a list of numbers and returns a new list containing only the even numbers.

```
def filter_even_numbers(lst):
    return [x for x in lst if x % 2 == 0]
```

Write a function that flattens a nested list of lists.

```
def flatten_list(nested_lst):
    return [item for sublist in nested_lst for item in sublist]
```

Write a function that takes a list of numbers and returns the second largest number.

```
def second_largest(lst):
    return sorted(list(set(lst)))[-2]
```

Level 3:

Write a function that takes a list and returns all possible subsets of that list.

```
from itertools import combinations
def all_subsets(lst):
    subsets = []
    for i in range(len(lst)+1):
        subsets.extend(list(combinations(lst, i)))
    return subsets
```

Write a function that takes a list and returns the list rotated to the right by n places.

```
def rotate_list(lst, n):
    return lst[-n:] + lst[:-n]
```

Write a function that takes a list of numbers and returns True if the list is sorted in ascending order, otherwise False.

```
def is_sorted(lst):
    return lst == sorted(lst)
```

Write a function that returns the most frequent element in a list. If there are ties, return any one of the most frequent elements.

```
from collections import Counter
def most_frequent(lst):
    return Counter(lst).most_common(1)[0][0]
```

Write a function that takes two lists and returns True if they are permutations of each other.

```
def are_permutations(lst1, lst2):
    return sorted(lst1) == sorted(lst2)
```

Previous :

Assignment - Questions :

Level 1

1. Write a for loop to print all even numbers between 1 and 20.
2. Write a for loop to print numbers between 30 and 50 that are divisible by 5.
3. Write a for loop to print the square of each number from 1 to 10.
4. Write a while loop to print all odd numbers between 1 and 15.
5. Write a while loop to print numbers from 100 down to 50 that are divisible by 10.
6. Write a for loop to print all prime numbers between 1 and 50.
7. Write a while loop to print the first 10 numbers of the Fibonacci sequence.
8. Write a for loop to print the multiplication table of 7.
9. Write a while loop to keep doubling a number until it exceeds 1000, starting from 1.
10. Write a for loop to print numbers between 1 and 20, but skip numbers divisible by 3.

Level 2

1. Write a for loop to print all numbers between 1 and 50, but stop the loop when you encounter a number divisible by both 8 and 9.
2. Write a while loop to calculate the sum of all numbers between 1 and 100 that are divisible by 4, and print the result.
3. Write a for loop to print all the digits of a given number in reverse order (e.g., for 1234, print 4 3 2 1).
4. Write a while loop to find the smallest number greater than 500 that is divisible by both 7 and 13.
5. Write a for loop to print the first 10 terms of the arithmetic sequence starting with 5, with a common difference of 3 (i.e., 5, 8, 11, ...).
6. Write a while loop to find the factorial of a given number, and print the result.

END