PES UNIVERSITY, BENGALURU

Perseverance | Excellence | Service

UE25CS151A – PYTHON FOR COMPUTATIONAL PROBLEM SOLVING LAB MANUAL

WEEK 5

TOPICS:

Programs on List and Tuples

OBEJCTIVE:

Solve practical problems using lists and tuples to manage collections of data.

Problem Statement 1: [List]

Given a list of integers nums and an integer target, find the indices of the two numbers such that they add up to target.

Example:

nums = [2, 7, 11, 15]

target = 9

Output: [0,1]

Solution:

```
# --- Input Data ---
nums = [2, 7, 11, 15,12,1]
target = 12
result_indices = []
# Iterate through each element with its index i
for i in range(len(nums)):
    # Iterate through the rest of the list starting from i + 1
    for j in range(i + 1, len(nums)):
        # Check if the two numbers sum up to the target
        if nums[i] + nums[j] == target:
            result_indices = [i, j]
            break  # Exit inner loop once found
print(f"Indices: {result_indices}")
```

```
Indices: [2, 5]
```

Problem Statement 2:[List]

A teacher has two equal groups of students. The first group is seated in the first half of a row, and the second group in the second half of the row. The teacher now wants to rearrange the seating so that students from both groups sit alternately.
Write a Python program to read the roll numbers of 2n students (entered in a single line separated by spaces) and rearrange them in an alternating order.

- The first n roll numbers represent Group A.
- The next n roll numbers represent Group B.
- The output should display the new arrangement as [A1, B1, A2, B2, ..., An, Bn].

```
Enter roll numbers separated by space: 101 102 103 201 202 203
Enter n: 3
Output: [101, 201, 102, 202, 103, 203]
```

Solution:

```
nums = input("Enter numbers separated by space: ").split()
for i in range(len(nums)):
    nums[i] = int(nums[i])
n = int(input("Enter n: "))
result = []
for i in range(n):
    result.append(nums[i])
    result.append(nums[i+n])
print(result)
```

```
Enter numbers separated by space: 1 2 3 4 5 6 7 8
Enter n: 4
[1, 5, 2, 6, 3, 7, 4, 8]
```

Problem Statement 3:[List]

Your goal is to write a program that combines two lists into a single new list. The new list should be created by taking elements from the two input lists in an alternating pattern, starting with the first element of the first list.
Once all the elements from the shorter list have been used, append the rest of the elements from the longer list to the end of the new list.

Input

Two lists, which may have different lengths. For example:

- List 1: m = [11, 22, 33]
- List 2: n = [66, 55, 77, 44, 88]

Output

A single new list containing the merged elements. Following the example above, the process would be:

1. Take 11 from m.
2. Take 66 from n.
3. Take 22 from m.
4. Take 55 from n.

5. Take 33 from m.
6. Take 77 from n.
7. List m is now empty. Append the rest of list n ([44, 88]).

The final output should be: [11, 66, 22, 55, 33, 77, 44, 88]

Solution:

```
m = [11, 22, 33]
n = [66, 55, 77, 44, 88]
# 1. Initialize an empty list to store the result
merged_list = []
# 2. Find the length of the shorter list
min_len = min(len(m), len(n))
# 3. Loop through the lists for the alternating part
for i in range(min_len):
    merged_list.append(m[i])
    merged_list.append(n[i])
# 4. Append the remaining elements from the longer list
if len(m) > len(n):
    # If m is longer, add its remaining elements
    merged_list.extend(m[min_len:])
elif len(n) > len(m):
    # If n is longer, add its remaining elements
    merged_list.extend(n[min_len:])
# 5. Print the final merged list
print(merged_list)
```

```
[11, 66, 22, 55, 33, 77, 44, 88]
```

Problem Statement 4: (Tuple)

A tuple is used to represent a shopping cart in an online grocery store.

cart = ("Milk", "Bread", "Butter", "Cheese", "Jam", "Eggs")

Tasks:

1. Pack the above items into a tuple. Unpack the first three items into separate variables and print them. Print the remaining items as a list and check its type.
2. Display all items in the cart using a loop.
3. Print only the first 3 items and the last 2 items using slicing.
4. Check whether **"Butter" and "Honey"** are present in the cart.
5. Add a new tuple **("Juice", "Fruits")** to the cart and display the updated cart.
6. Find the index of "**Cheese**" in the cart.
7. Count how many times "**Butter**" appears in the cart.

```
# Packing: items stored in a tuple
cart = ("Milk", "Bread", "Butter", "Cheese", "Jam", "Eggs", "Butter")
```

```
# 1. Unpacking the first three items
item1, item2, item3, *rest = cart      #* is a feature called extended
iterable unpacking.remaining items as list
print("First three unpacked items:", item1, item2, item3)
print(rest,type(rest))
# 2. Traversal
print("\nItems in cart:")
for item in cart:
    print(item)
# 3. Slicing
print("\nFirst 3 items:", cart[:3])
print("Last 2 items:", cart[-2:])
# 4. Membership operator
print("\nIs Butter in cart?", "Butter" in cart)
print("Is Honey in cart?", "Honey" in cart)
# 5. Concatenation
new_items = ("Juice", "Fruits")
updated_cart = cart + new_items
print("\nUpdated Cart:", updated_cart)
# 6. Index & Count
print("\nIndex of 'Cheese':", cart.index("Cheese"))
print("Count of 'Butter':", cart.count("Butter"))
```

```
First three unpacked items: Milk Bread Butter
['Cheese', 'Jam', 'Eggs', 'Butter'] <class 'list'>

Items in cart:
Milk
Bread
Butter
Cheese
Jam
Eggs
Butter

First 3 items: ('Milk', 'Bread', 'Butter')
Last 2 items: ('Eggs', 'Butter')

Is Butter in cart? True
Is Honey in cart? False

Updated Cart: ('Milk', 'Bread', 'Butter', 'Cheese', 'Jam', 'Eggs', 'Butter', 'Juice', 'Fruits')

Index of 'Cheese': 3
Count of 'Butter': 2
```

Problem Statement 5: [List]

Write a python program to remove the duplicate elements in the sorted list.

```
nums = [0, 0, 1, 1, 1, 2, 2, 3, 3, 4]
if not nums:
    k = 0
else:
    k = 1
    for i in range(1, len(nums)):
        if nums[i] != nums[i-1]:
            nums[k] = nums[i]
            k += 1
print(f"Number of unique elements (k): {k}")
print(f"Modified array: {nums[:k]}")
```

```
Number of unique elements (k): 5
Modified array: [0, 1, 2, 3, 4]
```

Problem Statement 6: (List and Tuple)

You are given a list of integers. Your goal is to compute the running sum of this list, but with a specific output format.
Write a program that processes the input list and returns a **list of tuples**. Each tuple in the result should contain two elements:

1. The **original number** from the input list.
2. The **running sum** at that number's position.

Input
A list of integers.

- nums = [1, 2, 3, 4]

Output
A list of tuples, where each tuple is a (number, running_sum) pair.

- [(1, 1), (2, 3), (3, 6), (4, 10)]

Example Walkthrough
Given the input [1, 2, 3, 4]:

- For the first element 1, the running sum is 1. The pair is (1, 1).
- For the second element 2, the running sum is 1 + 2 = 3. The pair is (2, 3).
- For the third element 3, the running sum is 3 + 3 = 6. The pair is (3, 6).
- For the fourth element 4, the running sum is 6 + 4 = 10. The pair is (4, 10).

Solution:

```
input_nums = [1, 2, 3, 4]
# 1. Initialize an empty list for the results and a running sum.
result_list = []
current_sum = 0
for num in input_nums:
    # 3. Add the current number to the running total.
```

```
    current_sum += num
    # 4. Create an immutable tuple of the (number, running_sum) pair.
    record = (num, current_sum)
    # 5. Append the new tuple to our result list.
    result_list.append(record)
# --- Final Output ---
print(f"Original list: {input_nums}")
print(f"List of running sum tuples: {result_list}")
```

```
Original list: [1, 2, 3, 4]
List of running sum tuples: [(1, 1), (2, 3), (3, 6), (4, 10)]
```

Practice Programs:

1. Create a list and fill it with numbers from n to 1 and then to 2 to n
   example: if n = 4, then list is [4, 3, 2, 1, 2, 3, 4]
2. Write a Python program that:
   Accepts a list of numbers as input (the list should be entered inside square brackets).
   Finds the leftmost even number in the list.
   If no even number exists, display a message "There exist no even numbers in the list."

   Example:
   Enter a list of numbers enclosed by square brackets: [11, 25, 39, 42, 57]
   The leftmost even number is: 42