**Assignment No. :1B**

**Aim -**

Using python perform List Operations and list slicing in detail with necessary programs.

**Theory-**

n Python, a list is a built-in data structure that stores an ordered collection of items, which can be of any type, including other lists. Lists are mutable, meaning their elements can be modified, and they support various operations such as indexing, slicing, appending, inserting, removing, sorting, reversing, and comprehensions. This makes them a flexible and dynamic choice for managing and manipulating data.

**Algorithm-**

**1. Create a List**

Step-1: Initialize an empty list.

Step-2: Initialize a list with elements.

### **2. Access Elements**

Step-1: Access an element by index.

Step-2: Access an element by negative index.

Step-3: Slice the list.

### **3. Modify the List**

Step-1: Change an element at a specific index.

Step-2: Add an element to the end of the list.

Step-3: Insert an element at a specific index.

Step-4: Extend the list by adding elements from another iterable.

Step-5: Remove an element by value.

Step-6: Remove an element by index.

Step-7:Rome last element

### **4. List Operations**

### Step-1:Concatenate: new\_list = list1 + list2

SteRepeat: repeated\_list = list \* n

### **5. Search and Find**

* Check Presence: exists = item in mylist
* Find Index: index = mylist.index(item)

### **6. Sort and Reverse**

* Sort Ascending: mylist.sort()
* Sort Descending: mylist.sort(reverse=True)
* Reverse: mylist.reverse()

### **7. List Comprehensions**

* Basic: new\_list = [expression for item in iterable]
* With Condition: new\_list = [expression for item in iterable if condition]

### **8. Nested Lists**

* Access Nested: element = mylist[i][j]
* Flatten: flat\_list = [elem for sublist in mylist for elem in sublist]

### **9. Clear and Delete**

* Clear: mylist.clear()
* Delete by Index: del mylist[index]
* Delete List: del mylist

### **10. Length and Iteration**

* Length: length = len(mylist)
* Iterate: for item in mylist:

**Datasets :**

1.list: This is the main list that the user interacts with. It is initially empty but can be populated with numbers through user input. Users can perform various operations on this list, such as adding, removing, or accessing elements.

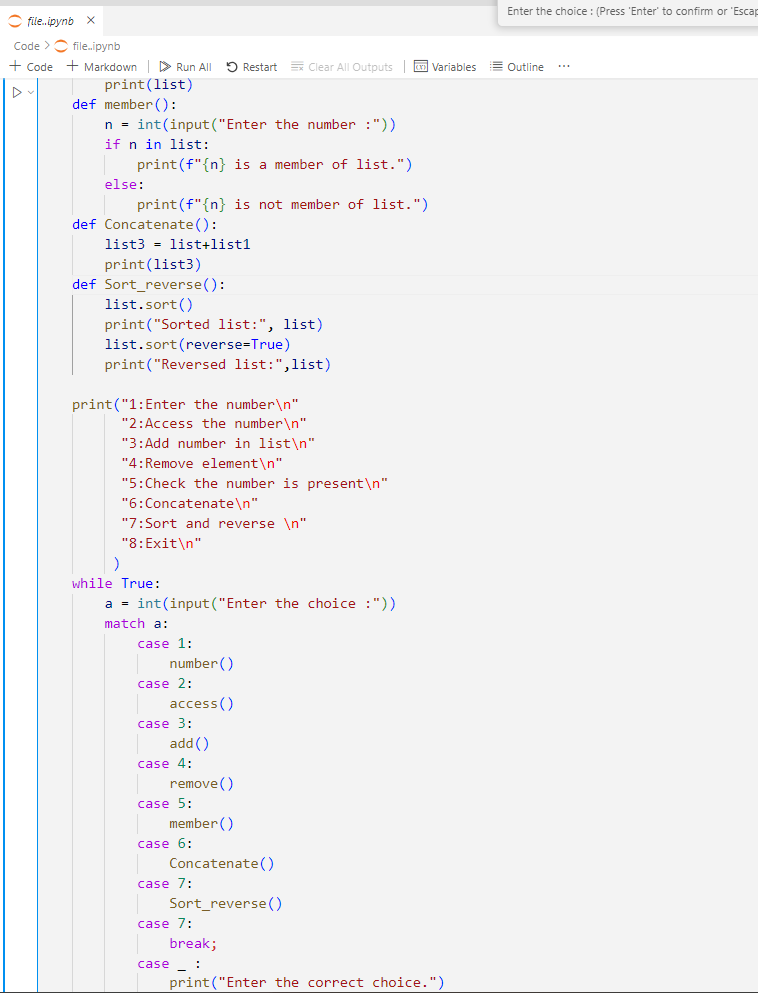
2.list1: This is a predefined list with the elements [4, 5, 6]. It is used for concatenation with the main list to create a new list (list3) when the Concatenate() function is called.

**Conclusion:**

This Python script provides a menu-driven interface for managing a list with various functionalities: adding, accessing, removing elements, and checking membership. It also allows concatenating with another list and sorting/reversing the list. The script uses a while loop and a match statement to handle user choices and perform the corresponding operations, with basic error handling for invalid inputs.

**Code:**





**Output:**

