**ASSIGNMENT- DAY 2- 11/03/2025**

**1. Perform Crud operation in an array.(Create, read, update, delete)**

class MyArray:

def \_\_init\_\_(self, size):

self.size = size

self.data = [None] \* size

def insert(self, index, value):

if 0 <= index < self.size:

self.data[index] = value

else:

print("Index out of range")

def get(self, index):

if 0 <= index < self.size:

return self.data[index]

**Output:**

[10, 20, 30, None, None]

20

[10, 99, 30, None, None]

[10, None, 30, None, None]

else:

print("Index out of range")

return None

def update(self, index, value):

if 0 <= index < self.size:

self.data[index] = value

else:

print("Index out of range")

def delete(self, index):

if 0 <= index < self.size:

self.data[index] = None

else:

print("Index out of range")

def display(self):

print(self.data)

arr = MyArray(5)

arr.insert(0, 10)

arr.insert(1, 20)

arr.insert(2, 30)

arr.display()

print(arr.get(1))

arr.update(1, 99)

arr.display()

arr.delete(1)

arr.display()

**2: Take user input of numbers to perform linear search in an array or list.**

def linear\_search(arr, key):

**Output:**

Enter the number of elements: 2

Enter the elements: 1

4

Enter the number to search: 5

Element 5 not found

Enter the number of elements: 4

Enter the elements: 1

4

8

6

Enter the number to search: 4

Element 4 found at 1

for i in range(len(arr)):

if arr[i] == key:

return i

return -1

n = int(input("Enter the number of elements: "))

arr = []

print("Enter the elements:")

for i in range(n):

arr.append(int(input()))

key = int(input("Enter the number to search: "))

result = linear\_search(arr, key)

if result != -1:

print(f"Element {key} found at index {result}")

else:

print(f"Element {key} not found")

**Exercises:**

**1. Basic array operations like setting values, getting values, displaying the array, and searching for elements.**

class MyArray:

def \_\_init\_\_(self, size):

self.size = size

self.data = [None] \* size

def setvalues(self, index, value):

if 0 <= index < self.size:

self.data[index] = value

else:

print("Index out of range")

def getvalues(self, index):

if 0 <= index < self.size:

**Output:**

[56, 78, 60, 23, 45]

2

-1

return self.data[index]

else:

print("Index out of range")

return None

def display(self):

print(self.data)

def search(self, value):

if value in self.data:

return self.data.index(value)

else:

return -1

ar = MyArray(5)

ar.setvalues(0, 56)

ar.setvalues(1, 78)

ar.setvalues(2, 90)

ar.setvalues(3, 23)

ar.setvalues(4, 45)

ar.display()

print(ar.search(90))

print(ar.search(100))

**2. Stack operations with push, pop, display and isEmpty.**

class Stack:

def \_\_init\_\_(self):

self.stack = []

def push(self,item):

self.stack.append(item)

def pop(self):

if not self.is\_empty():

return self.stack.pop()

else:

print('Stack is empty')

return None

**Output:**

[1, True, 3]

[1, True]

[1]

def display(self):

print(self.stack)

def is\_empty(self):

return len(self.stack) == 0

Stack1 = Stack()

Stack1.push(1)

Stack1.push(True)

Stack1.push(3)

Stack1.display()

Stack1.pop()

Stack1.display()

Stack1.pop()

Stack1.display()