LAB 2: CMPS347

|  |  |
| --- | --- |
| **Beirut Arab University Faculty of Science**  **Mathematics and Computer Science Department** | **Course: Data Structure Semester: Fall 2024-2025**  **Lab Sheet # 2** |

# Question 1:

# We have a special scenario where the array can only be interacted with from the 'top' (the first element). When we want to add a new number, we must place it at the top of the array. Use the end of the array to represent the top of the array. Similarly, when removing a number, we can only remove the one at the top.

# If we want to work with any element that is not at the top, we need to first remove the elements above it temporarily. After accessing the desired element, we will need to carefully put the removed elements back in their original order, as we don't want to disrupt the array.

# Hint: Consider using a temporary array to store the removed elements, so nothing gets lost in the process.

Add to the class ArrayInt the following methods:

|  |
| --- |
| **ArrayInt** |
| * **a: int[]** * **top: int** * **size:int** |
| * **ArrayInt( size: int)** * **isEmpty(): boolean** * **isFull() : Boolean** * **InsertElementAtTop(value: int): void** * **DeleteFromTop():int** * **Display(): void** * **findUniqueValues(): void** * **ReverseArray(): ArrayInt** * **PrimeArray():void** * **InsertElementAtStart(value: int): void //Note that you can only insert at top** |

**Attributes** of class **ArrayInt**

* **a**: an array where numbers are placed
* **top:** number of elements in the array

**Methods** of class **ArrayInt:**

* **ArrayInt():** constructor that initializes the class attributes
* **isEmpty():** returns true if the array is empty and false otherwise.
* **isFull():** returns true if the array is full and false otherwise.
* **InsertAtTop(int):** adds an element at the top of the array.

**Hint:** Increment the variable Top so that it can now reference to the next memory location.

A diagram of a number

Description automatically generated

* Add element at the position of incremented top. This is referred to as adding new element at the top of the array.
* **DeleteFromTop():** Deletes an element from the top of the array.

Hint: The value of the variable top will be incremented by 1 whenever an item is deleted from the stack. The top most element of the stack is stored in an another variable and then the top is decremented by 1. the operation returns the deleted value that was stored in another variable as the result.

* **Display():** prints the elements of the array.
* **FindUniqueValues ():** Finds and displays unique values in the array.
* **ReverseArray**(): returns a new array where the elements are reversed while keeping the original array as it is
* **PrimeArray**()displays all the prime elements in the array.

**Write the main program to test all methods of the class ArrayInt**

**Question 2:**

We have a unique situation where the array can only be accessed from the beginning and the end of the array. Here's how it works:

* **Inserting Elements**: When we insert a new number, we add it to the end of the array. This is like adding a new item to the back of a line.
* **Removing Elements**: When we remove an element, we take it from the bottom of the array. This is like serving the first person in line.
* **Accessing Elements**: If you want to access elements other than the bottom one, you need to temporarily remove all elements up to the one you want to access. You should store these removed elements temporarily and then put them back to preserve the original order.

### **Hint:**Use a temporary array to keep track of the removed elements so you can restore them afterward.

|  |
| --- |
| **ArrayInt** |
| * **a: int[]** * **size:int** * **front:int** * **rear:int** |
| * **ArrayInt( size: int)** * **isEmpty(): boolean** * **isFull() : Boolean** * **insertElementAtTop(value: int): void** * **DeleteFromBottom():void** * **Display(): void** * **SearchForElement(value:int): int** * **DeleteElement(element:int):void** * **maxEven():int** |

* **ArrayInt():** constructor that initializes the class attributes
* **isEmpty():** returns true if the array is empty and false otherwise.
* **isFull():** returns true if the array is full and false otherwise.
* **insertElementAtTop (int):** adds an element at the end of the array.
* **DeleteFromBottom():** Deletes an element from the front of the array.
* **Display():** prints the elements of the array.
* **SearchForElement(element:int)**: returns the index of the element.
* **DeleteElement(value:int)** deletes the entered value from the array.
* **maxEven** (): returns the maximum even number in the array.

**Write the main program to test all methods of the class ArrayInt**