

# **Data Structure Lab**



## **Lab # 03**

### **Array based List**

Instructor: Muhammad Saood Sarwar

Email: [saood.sarwar@nu.edu.pk](mailto:saood.sarwar@nu.edu.pk)

Course Code: CL2001

Department of Computer Science,  
National University of Computer and Emerging Sciences FAST Peshawar  
Campus

<b>Array Based List</b>	<b>3</b>
<b>1. insert:</b>	<b>3</b>
<b>2. insertAt:</b>	<b>4</b>
<b>3. deleteElement:</b>	<b>4</b>
<b>4. sortList:</b>	<b>5</b>
<b>5. reverseList:</b>	<b>5</b>

## Array Based List

Array based list is like a list but lists in python and arrays in C++ are not the same. Arrays are fixed size whereas lists are not fixed size.

To access the elements of an array we can either use indexes or pointer(s).

Pointer stores the address of memory location and that's how we are going to access the memory location which holds the data of our array.

Because the arrays are contiguous, the pointer just needs to point to the 0th index of the array. It will mean it's pointing to the cells of the array.

We will be discussing 6 major functions:

1. insert
2. insertAt
3. deleteElement
4. sortList
5. reverseList

### 1. insert:

In the insert function we need to insert the value at the current free slot. Let's say we have an array of 10 elements and out of 10 only the first 5 elements are filled up.

E.g.

Array = [1,5,9,10,45,0,0,0,0,0]

0 values show that these are empty locations. Let's say we want to insert a new value, 88, in the current position so the array will become like this:

Array = [1,5,9,10,45, 88,0,0,0,0]

### Pseudo code:

**insert(x)**

if !isFull()

    Array[i] ← x

    length ← length + 1

else

    No Free Slot Available

## 2. insertAt:

This function will insert the data passed to it in the memory location that is passed to it.

E.g.

Array = [1,5,9,10,45, 88,0,0,0,0]

If we call insertAt(x,y), we pass it data (x) and position (y), it inserts the data in position that is passed to it.

insertAt(17,3)

Array = [1,5,9,17,10,45, 88,0,0,0]

**Pseudo code:**

**insertAt(x, loc)**

If valid index:

Shift array to the right one index from the desired index(where we need to insert data)

Add data to the free index

Increment length

## 3. deleteElement:

This function will remove data from the memory location whose value is passed to it, i.e. the memory location (slot) will be freed.

E.g.

If we call deleteElement(9) it will search for 9 in the array and remove the data containing 9.

Array = [1,5,9,10,45, 88,0,0,0,0]

Updated array:

Array = [1,5,10,45,88, 0,0,0,0,0]

**Pseudo code:**

**deleteElement(x)**

if x found in arr:

Shift array to the left one index from the desired index(where we

need to delete data)

Decrement length

#### **4. sortList:**

This function would sort the list in ascending order. You could use any sorting Algorithm.

#### **5. reverseList:**

This function would reverse the list.

#### **Pseudo code:**

##### **reverseList()**

Iterate a loop till half of list size

Swap the first index value with the last index value. Similarly,

Second index value with the Second last index value respectively

...

---