**Exercise 4 : Employee Management System**

**Explain how arrays are represented in memory and their advantages.**

 **Contiguous Memory Allocation:** Arrays allocate a contiguous block of memory for storing elements. This means all elements are stored in consecutive memory locations, which allows for efficient indexing.

 Arrays use indices to access elements. Each index points to a specific memory location, which allows for O(1) time complexity for accessing elements by index.

**Advantages :**

 **Fast Access:** Direct indexing provides O(1) access time to elements.

 **Simplicity:** Arrays are straightforward to implement and use.

 **Memory Efficiency:** Arrays use a minimal amount of additional memory beyond the storage of elements themselves.

**Analyze the time complexity of each operation (add, search, traverse, delete).**

**Add Operation:**

* Adding an element involves placing it in the next available position, which is an O(1) operation. However, if the array is full, resizing is required, which involves creating a new array and copying elements, making it O(n) in such cases
* Complexity is O(1) if there is space in the array.

**Search Operation:**

* Searching for an element requires scanning through the array until the desired element is found or the end of the array is reached.
* Complexity: O(n)

**Traverse Operation:**

* Traversing involves visiting each element once, leading to linear time complexity.
* Complexity: O(n)

**Delete Operation:**

* Deleting an element involves finding it first (O(n)) and then shifting all subsequent elements to fill the gap (O(n)).
* Complexity: O(n)

**Discuss the limitations of arrays and when to use them.**

The limitations of the array include :

* Arrays have a fixed size, which means you need to know the capacity in advance. If the capacity is exceeded, you need to resize the array, which is inefficient.
* Adding or removing elements, especially in the middle of the array, requires shifting elements, which can be costly.
* Arrays cannot grow or shrink dynamically, which makes them less flexible compared to other data structures.

Array can be used when :

* + the number of elements is known in advance and does not change frequently.
  + Direct access to elements by index is needed for performance reasons.
  + Memory overhead needs to be minimized, as arrays have minimal additional memory usage beyond storing the elements.