**Exercise 4: Employee Management System**

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

Arrays are stored in contiguous blocks of memory, which means each element is located next to its neighbour. This makes accessing elements by index very fast (O(1) time complexity). Arrays use zero-based indexing, which means the first element is at index 0, the second at index 1, and so on. The size of an array is fixed at the time of its creation and cannot be changed dynamically.

Advantages:

* Constant-time access to elements using an index.
* Minimal overhead since elements are stored in contiguous memory locations.
* Contiguous memory allocation improves cache performance.

**ANALYSIS**

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

Add Operation:

* Best Case: O(1) - When adding to the first available position.
* Worst Case: O(1) - Always O(1) since adding is direct as long as there is space.

Search Operation:

* Best Case: O(1) - When the element to be searched is the first element.
* Average Case: O(n) - On average, the element is somewhere in the middle.
* Worst Case: O(n) - When the element is the last or not present.

Traverse Operation:

* Best Case: O(n) - Always O(n) since all elements need to be visited.
* Worst Case: O(n) - Always O(n) since all elements need to be visited.

Delete Operation:

* Best Case: O(1) - When the element to be deleted is the last element.
* Average Case: O(n) - Requires shifting elements after the deleted element.
* Worst Case: O(n) - When the element to be deleted is the first element, requiring shifting of all subsequent elements.

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

* Once an array is created, its size cannot be changed. This can lead to wasted space or the need to create a larger array and copy elements if more space is needed.
* Inserting or deleting elements, especially in the middle of the array, can be inefficient as it requires shifting elements.

Arrays are best used when the number of elements is known in advance and does not change frequently, and fast access to elements is required.