**Exercise 4: Employee Management System**

**1. Understand Array Representation**

Arrays are **contiguous blocks of memory** where each element is stored at a fixed distance from the previous one. This allows **constant-time access (O(1))** to any element using its index. Arrays are efficient for storing fixed-size collections and offer fast traversal.

**Advantages of arrays:**

* Fast access to elements using indices.
* Simple to implement.
* Low memory overhead compared to complex structures.

However, they have a **fixed size** once created and require shifting elements during insertions or deletions.

**2. Analysis**

* **Add Operation:** Adding an employee takes **O(1)** time as the employee is inserted at the next available index.
* **Search Operation:** Searching by employee ID requires scanning through the array, so the time complexity is **O(n)**.
* **Traverse Operation:** Displaying all employees requires visiting each element, resulting in **O(n)** time.
* **Delete Operation:** Deleting an employee involves finding the employee (O(n)) and then shifting all subsequent elements one position left, so overall time complexity is **O(n)**.

**Limitations of Arrays**

Arrays have fixed size, so they can’t dynamically grow or shrink. When you don’t know the number of elements in advance or need frequent insertions/deletions, arrays become inefficient. In such cases, **ArrayList** or **LinkedList** is preferred.

Arrays are best used when:

* The number of records is known and fixed.
* Fast index-based access is required.
* Memory usage must be minimal.

**3. Outputs**

