**ANSWER**

**Exercise 4: Employee Management System**

**Scenario:**

You are developing an employee management system for a company. Efficiently managing employee records is crucial.

**Steps:**

1. **Understand Array Representation:**
   * Explain how arrays are represented in memory and their advantages.

Arrays are a fundamental data structure in programming, represented in memory as a contiguous block of elements, each accessed via an index. Here's how they are typically represented and their advantages:

* **Memory Representation**: Arrays are stored in a contiguous block of memory, meaning that each element is positioned next to its neighbors. The base address (the address of the first element) allows for constant-time access to any element by calculating its memory address using the formula:

Address of a[i]=Base Address+i×Size of each element\text{Address of } a[i] = \text{Base Address} + i \times \text{Size of each element}Address of a[i]=Base Address+i×Size of each element

* **Advantages**:
  + **Constant-Time Access (O(1))**: Any element can be accessed directly via its index.
  + **Efficient Traversal**: Iterating through an array is straightforward and efficient.
  + **Memory Locality**: Contiguous memory allocation improves cache performance, leading to faster data access.

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

**Add Employee**:

* Time Complexity: O(1)

**Search Employee**:

* Time Complexity: O(n)

**Traverse Employees**:

* Time Complexity: O(n)

**Delete Employee**:

* Time Complexity: O(n)
  + Discuss the limitations of arrays and when to use them.

**Fixed Size**: Arrays have a fixed size, which must be defined at creation. This can lead to wasted space if the array is too large or the need for resizing if it is too small.

**Insertion and Deletion**: While accessing elements is efficient, insertion and deletion can be inefficient (O(n)) due to the need to shift elements.