**Exercise 4: Employee Management System**

**Scenario:**

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

**Understand Array Representation**

**How Arrays Are Represented in Memory**

* Arrays in Java are **contiguous memory blocks**.
* Each element is placed next to the other in memory, enabling **constant-time access (O(1))** using an index.
* Java arrays are **fixed-size**, meaning you must define the number of elements upfront.

**Advantages of Arrays**

* **Fast element access** by index: O(1)
* **Simple** to use and implement
* Good for **storing homogeneous data types**

**Analysis**

**Time Complexity**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Time Complexity** | **Explanation** |
| Add Employee | O(1) | Insert at end (if space exists) |
| Search | O(n) | Linear search by ID |
| Traverse | O(n) | Iterate all employees |
| Delete | O(n) | Linear search + shift |

**Limitations of Arrays**

* **Fixed size** — can't dynamically expand when full.
* **Inefficient deletion** — requires shifting elements.
* No built-in sorting/searching for objects — must be implemented manually.
* **Best used** when:
  + Maximum size is known in advance.
  + Records are stable (few insert/delete operations).
  + Fast index-based access is required.

For flexible and scalable employee data management, **ArrayList** or **HashMap** would be better long-term solutions.