**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.
2. **Setup:**
   * Create a class Employee with attributes like **employeeId**, **name**, **position**, and **salary**.
3. **Implementation:**
   * Use an array to store employee records.
   * Implement methods to **add**, **search**, **traverse**, and **delete** employees in the array.
4. **Analysis:**
   * Analyze the time complexity of each operation (add, search, traverse, delete).
   * Discuss the limitations of arrays and when to use them.

**CODE:**

using System;

class Employee

{

public int EmployeeId { get; set; }

public string Name { get; set; }

public string Position { get; set; }

public decimal Salary { get; set; }

public Employee(int id, string name, string position, decimal salary)

{

EmployeeId = id;

Name = name;

Position = position;

Salary = salary;

}

}

class EmployeeManager

{

private Employee[] employees = new Employee[100];

private int count = 0;

public void AddEmployee(Employee emp)

{

if (count < employees.Length)

employees[count++] = emp;

}

public Employee SearchEmployee(int id)

{

for (int i = 0; i < count; i++)

if (employees[i].EmployeeId == id)

return employees[i];

return null;

}

public void TraverseEmployees()

{

for (int i = 0; i < count; i++)

{

var emp = employees[i];

Console.WriteLine($"{emp.EmployeeId}: {emp.Name}, {emp.Position}, Salary: {emp.Salary}");

}

}

public void DeleteEmployee(int id)

{

for (int i = 0; i < count; i++)

{

if (employees[i].EmployeeId == id)

{

for (int j = i; j < count - 1; j++)

employees[j] = employees[j + 1];

employees[--count] = null;

break;

}

}

}

}

class Program4

{

static void Main()

{

var manager = new EmployeeManager();

manager.AddEmployee(new Employee(101, "John", "Manager", 75000));

manager.AddEmployee(new Employee(102, "Jane", "Developer", 60000));

manager.TraverseEmployees();

Console.WriteLine("\nSearching for ID 102: " + manager.SearchEmployee(102)?.Name);

manager.DeleteEmployee(101);

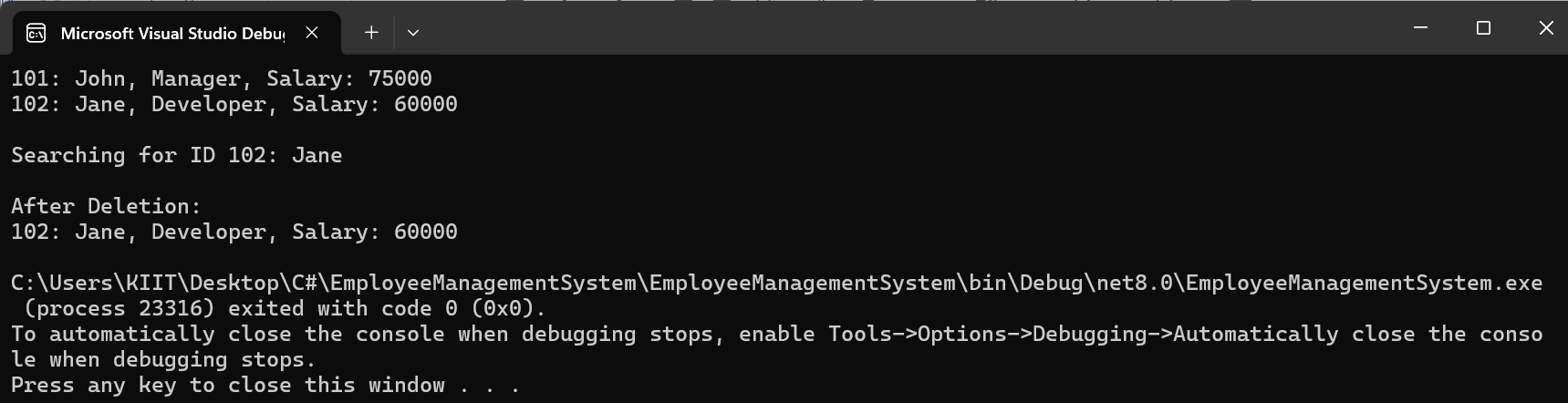
Console.WriteLine("\nAfter Deletion:");

manager.TraverseEmployees();

}

}

**OUTPUT:**



**ANALYSIS:**

**Time Complexities:**

* Add: O(1)
* Search: O(n)
* Traverse: O(n)
* Delete: O(n) (shifting needed)

**Limitations of Arrays:**  
Fixed size, inefficient insertion/deletion. Best for static or small datasets.