

# Set 5: Dynamic Memory Allocation & Structure Pointer

## Instructions

This set of questions focuses on the use of dynamic memory allocation and pointers with structures

## Deadline:

20 November 2024

## Question 1: Online Shopping Cart System

*Scenario:* You are developing an online shopping cart for a website. Customers can add a variable number of items to their cart, each with a price and quantity. After adding all the items, the total cost should be calculated.

- Dynamically allocate memory for **n** items (where **n** is provided by the user).
- For each item, store the price and quantity.
- Compute the total cost by multiplying the price by the quantity for each item.
- Display the total cost of the cart.

**Edge Case:** If the user enters 0 items, print a message saying the cart is empty and no total can be calculated.

**Expected Output:**

- Enter the number of items in your shopping cart: 3
- Enter price for item 1: 10.5
- Enter quantity for item 1: 2
- Enter price for item 2: 15.0
- Enter quantity for item 2: 1
- Enter price for item 3: 7.25
- Enter quantity for item 3: 4
- Total cost of your shopping cart: \$72.75

**Edge Case Output:**

- Enter the number of items in your shopping cart: 0
- Your shopping cart is empty. No total cost to calculate.

## Question 2: Student Grades and Average Calculation

*Scenario:* A teacher wants to calculate the average grade of  $n$  students. Each student's grade should be stored dynamically. After entering the grades, the system should compute and display the average grade of the class.

- Dynamically allocate memory for  $n$  students' grades.
- Take input for each student's grade and store it in the allocated memory.

- Calculate and display the average grade for the class.

**Edge Case:** If the number of students  $n$  is 0, display an appropriate message indicating no students to grade.

**Expected Output:**

- Enter the number of students: 4
- Enter grade for student 1: 85
- Enter grade for student 2: 90
- Enter grade for student 3: 78
- Enter grade for student 4: 92
- Average grade of the class: 86.25

**Edge Case Output:**

- Enter the number of students: 0
- No students in the class. Program exiting.

### Question 3: Queue Management for Customer Service

*Scenario:* You need to implement a queue system for a customer service center. Customers join the queue by entering their IDs, and they are served in a first-come, first-served order. The number of customers can vary, and the queue size should adjust dynamically.

- Dynamically allocate memory for storing the customer IDs.
- Implement the functionality to enqueue (add customers) and dequeue (serve customers).
- Keep track of the front and rear of the queue.

- Display the current state of the queue whenever requested.

**Edge Case:** If the queue is empty (i.e., no customers to serve), display a message saying the queue is empty.

**Expected Output:**

- Enter the maximum size of the queue: 5
- 1. Enqueue (Add customer)
- Enter customer ID: 1001
- 1. Enqueue (Add customer)
- Enter customer ID: 1002
- Current queue: 1001 1002
- Serving customer with ID: 1001

**Edge Case Output:**

- Enter your choice: 2
- Queue is empty. No customers to serve.

## Question 4: Hotel Room Reservation System

*Scenario:* A hotel needs to keep track of guest reservations. The program should store guest details (name and room number) for **n** guests, where **n** is entered by the user. After storing the reservations, the system should display all guest details.

- Dynamically allocate memory to store guest information (name and room number).
- After inputting the guest details, display all the reservations in a neat format.

- Free the allocated memory after use.

**Edge Case:** If the user enters 0 guests, display a message indicating that there are no reservations.

**Expected Output:**

- Enter the number of guest reservations: 2
- Enter name for guest 1: John
- Enter room number for guest 1: 101
- Enter name for guest 2: Alice
- Enter room number for guest 2: 102
- Current hotel reservations:
- Guest: John, Room Number: 101
- Guest: Alice, Room Number: 102

**Edge Case Output:**

- Enter the number of guest reservations: 0
- No reservations to make. Program exiting.

## Question 5: Library Book Catalog

*Scenario:* A library maintains a catalog of books with details such as title, author, and publication year. The program should dynamically allocate memory to store information for  $n$  books. After entering the book details, the program should display the list of all books in the catalog.

- Dynamically allocate memory for  $n$  books.
- For each book, store its title, author, and publication year.
- After storing the book details, display the entire catalog in a readable format.

**Edge Case:** If the number of books is 0, display a message saying no books were added to the catalog.

**Expected Output:**

- Enter the number of books in the library: 2
- Enter title for book 1: C Programming
- Enter author for book 1: Dennis Ritchie
- Enter year of publication for book 1: 1978
- Enter title for book 2: Data Structures
- Enter author for book 2: Mark Allen Weiss
- Enter year of publication for book 2: 2015
- Library Catalog:
- Title: C Programming, Author: Dennis Ritchie, Year: 1978
- Title: Data Structures, Author: Mark Allen Weiss, Year: 2015

**Edge Case Output:**

- Enter the number of books in the library: 0
- No books to add. Program exiting.

## Question 6: Employee Information System

*Scenario:* You are creating a system for storing and displaying employee information such as name, ID, and salary. The employee data is stored in a structure, and you need to use pointers to access and modify the data.

- Define a structure to store an employee's name, ID, and salary.
- Dynamically allocate memory for an array of  $n$  employees.
- Use structure pointers to input and display the details of each employee.

**Edge Case:** If the number of employees  $n$  is 0, print a message saying no employee data is available.

**Expected Output:**

- Enter the number of employees: 2
- Enter name for employee 1: Alice
- Enter ID for employee 1: 101
- Enter salary for employee 1: 50000.00
- Enter name for employee 2: Bob
- Enter ID for employee 2: 102
- Enter salary for employee 2: 55000.00
- Employee 1: Alice, ID: 101, Salary: 50000.00
- Employee 2: Bob, ID: 102, Salary: 55000.00

**Edge Case Output:**

- Enter the number of employees: 0
- No employee data to display.

## Question 7: Student Record System

*Scenario:* A school needs to store and manipulate student records, such as name, roll number, and marks. The student data is stored in a structure, and pointers will be used to modify and display each student's record.

- Define a structure to store student details: name, roll number, and marks.
- Dynamically allocate memory for  $n$  students.

- Use structure pointers to input and display each student's record.

**Edge Case:** If `n` is 0, print a message indicating no students to display.

**Expected Output:**

- Enter the number of students: 3
- Enter name for student 1: John
- Enter roll number for student 1: 1
- Enter marks for student 1: 85
- Enter name for student 2: Emma
- Enter roll number for student 2: 2
- Enter marks for student 2: 90
- Enter name for student 3: Lily
- Enter roll number for student 3: 3
- Enter marks for student 3: 95
- Student 1: John, Roll Number: 1, Marks: 85
- Student 2: Emma, Roll Number: 2, Marks: 90
- Student 3: Lily, Roll Number: 3, Marks: 95

**Edge Case Output:**

- Enter the number of students: 0
- No student records to display.



## Question 8: Book Information System

*Scenario:* You are designing a system to store and manage books in a library. Each book has a title, author, and ISBN number. Use structure pointers to dynamically store and access the book details.

- Define a structure to store the title, author, and ISBN number of a book.
- Dynamically allocate memory for  $n$  books.
- Use pointers to access and display each book's details.

**Edge Case:** If  $n$  is 0, print a message saying no books to display.

**Expected Output:**

- Enter the number of books: 2
- Enter title for book 1: C Programming
- Enter author for book 1: Dennis Ritchie
- Enter ISBN for book 1: 1234567890
- Enter title for book 2: Data Structures
- Enter author for book 2: Mark Weiss
- Enter ISBN for book 2: 0987654321
- Book 1: C Programming, Author: Dennis Ritchie, ISBN: 1234567890
- Book 2: Data Structures, Author: Mark Weiss, ISBN: 0987654321

**Edge Case Output:**

- Enter the number of books: 0
- No books in the library.

## Question 9: Inventory Management System

*Scenario:* You need to create a system for managing products in an inventory. Each product has a name, code, and price. Use structure pointers to store and display the details of each product.

- Define a structure for storing the product's name, code, and price.
- Dynamically allocate memory for  $n$  products.
- Use structure pointers to input and display the product details.

**Edge Case:** If  $n$  is 0, display a message saying no products are available in the inventory.

### Expected Output:

- Enter the number of products: 2
- Enter name for product 1: Laptop
- Enter product code for product 1: P12345
- Enter price for product 1: 800.50
- Enter name for product 2: Mobile Phone
- Enter product code for product 2: P67890
- Enter price for product 2: 500.25
- Product 1: Laptop, Code: P12345, Price: 800.50
- Product 2: Mobile Phone, Code: P67890, Price: 500.25

### Edge Case Output:

- Enter the number of products: 0
- No products to display.

## Question 10: Contact Management System

*Scenario:* You are building a contact management system to store information about contacts such as name, phone number, and email address. Use structure pointers to store and display the contact information.

- Define a structure for storing the contact's name, phone number, and email address.
- Dynamically allocate memory for `n` contacts.
- Use structure pointers to input and display the contact details.

**Edge Case:** If `n` is 0, print a message saying no contacts were added.

**Expected Output:**

- Enter the number of contacts: 2
- Enter name for contact 1: John Doe
- Enter phone number for contact 1: 9876543210
- Enter email for contact 1: johndoe@gmail.com
- Enter name for contact 2: Alice Smith
- Enter phone number for contact 2: 1234567890
- Enter email for contact 2: alicsmith@yahoo.com
- Contact 1: John Doe, Phone: 9876543210, Email: johndoe@gmail.com
- Contact 2: Alice Smith, Phone: 1234567890, Email: alicsmith@yahoo.com

**Edge Case Output:**

- Enter the number of contacts: 0
- No contacts to display.