Structures and Unions in C

1. Array of Structures

Q1: Store and Display Student Details

Write a C program to store details of multiple students, including their name, roll number, and marks, using an array of structures. Then, display all student details.

Example Input:

```
John, 101, 85
Alice, 102, 90
Bob, 103, 78
```

Example Output:

```
John (Roll No: 101) - Marks: 85
Alice (Roll No: 102) - Marks: 90
Bob (Roll No: 103) - Marks: 78
```

Q2: Find the Student with the Highest Marks

Modify the above program to find and display the student who has the highest marks.

Example Output:

```
Highest Scorer: Alice (Roll No: 102) - Marks: 90
```

Q3: Filter Employees by Salary

Create an `Employee` structure with fields for employee ID, name, and salary. Store multiple employee records in an array and display only those employees whose salary is greater than 50,000.

Example Input:

```
John, 1001, 45000
Alice, 1002, 60000
Bob, 1003, 55000
```

Example Output:

```
Alice (ID: 1002) - Salary: 60000
Bob (ID: 1003) - Salary: 55000
```

Q4: Sort Books by Price in Ascending Order

Define a `Book` structure with fields for title, author, and price. Store book details in an array and display them sorted by price in ascending order.

Example Input:

```
Book A - 500
Book B - 300
Book C - 700
```

Example Output:

```
Book B - 300
Book A - 500
Book C - 700
```

2. Pointer to Structure

Q5: Store and Display Car Details Using a Pointer

Write a program to store details of a car (brand, model, price) using a structure pointer and display the details.

Example Input:

```
Toyota, Corolla, 15000
```

Example Output:

```
Car: Toyota Corolla - Price: 15000
```

Q6: Store Product Details Dynamically Using a Pointer

Create a `Product` structure with fields for product ID, name, and cost. Use a pointer to for storing product details and display them.

Example Input:

```
101, Laptop, 75000
```

Example Output:

```
Product 101 - Laptop - Cost: 75000
```

Q7: Calculate the Area of a Rectangle Using a Pointer

Define a `Rectangle` structure with fields for length and width. Use a pointer to store values and compute the area.

Example Input:

```
Length: 10, Width: 5
```

Example Output:

```
Area: 50
```

Q8: Modify Person Details Using Pointers

Create a `Person` structure with fields for name and age. Modify the age value using a pointer.

Example Input:

```
Original: Alice, 25
Modified: Alice, 30
```

Example Output:

```
Updated Age: 30
```

3. Function and Structure

Q9: Take Student Details through a Function and Display Them

Write a function to take student details (name, roll number, marks) and return a structure. Display the details in the `main` function.

Example Input:

```
John, 101, 85
```

Example Output:

```
Student: John (Roll No: 101) - Marks: 85
```

Q10: Compare Two Employees' Salaries Using a Function

Write a function that takes two employee structures as arguments and compares their salaries. Display who earns more.

Example Input:

```
John, 45000
Alice, 60000
```

Example Output:

```
Alice has a higher salary.
```

Q11: Calculate the Distance Between Two Points

Define a `Point` structure with x and y coordinates. Create a function to calculate the distance between two points.

Example Input:

```
Point1: (2, 3)
Point2: (5, 7)
```

Example Output:

```
Distance: 5.0
```

Q12: Calculate Employee Bonus Based on Experience

Write a function to calculate an employee's bonus (20% of salary if experience is more than 10 years).

Example Input:

```
Alice, 60000, 12 years
```

Example Output:

```
Bonus: 12000
```

4. Nested Structure and Union

Q13: Store Student Date of Birth Using a Nested Structure

Define a nested structure `Student` with `DOB` as a substructure (day, month, year). Take input and display the birth date.

Example Input:

```
John, 15, 8, 2000
```

Example Output:

```
John - DOB: 15/08/2000
```

Q14: Store Book Publisher Details Using a Nested Structure

Define a `Book` structure with `Publisher` as a nested structure (name, year). Display book details.

Example Input:

```
Title: C Programming, Publisher: McGraw, 2020
```

Example Output:

```
C Programming - McGraw (2020)
```

Q15: Demonstrate Memory Sharing in a Union

Create a union `Data` that holds an integer, float, and character. Assign values and observe the last stored value.

Example Input:

```
int: 5, float: 4.2, char: 'A'
```

Example Output:

```
Only 'A' is retained due to memory sharing.
```

Q16: Use a Union for Vehicle Types

Create a `Vehicle` union that can store either a car model or a bike model but not both at the same time.

Example Input:

```
Vehicle: Car, Model: BMW
```

Example Output:

```
Vehicle Type: Car, Model: BMW
```

5. Typedef with Structure

Q17: Use Typedef for Employee Details

Use `typedef` to define an `Employee` structure and display employee details.

Example Input:

```
John, 101, 50000
```

Example Output:

```
Employee ID: 101 - John - Salary: 50000
```

Q18: Use Typedef for Circle Area Calculation

Define a `Circle` structure using `typedef` and calculate the area.

Example Input:

```
Radius: 7
```

Example Output:

```
Area: 153.94
```

Q19: Use Typedef for a Movie Database

Store and display a movie's title, director, and year using `typedef`.

Example Input:

```
Inception, Christopher Nolan, 2010
```

Example Output:

```
Movie: Inception - Director: Christopher Nolan - Released: 2010
```

Q20: Use Typedef for Time Addition

Create a `Time` structure using `typedef` and add two time values.

Example Input:

```
Time1: 2h 45m 30s
Time2: 1h 20m 40s
```

Example Output:

```
Total Time: 4h 6m 10s
```