Structure related problems
(Total # questions)

No.	Problem statement	Difficulty level
1	Declare a structure of students with three member variables (name, id and cgpa), where name is a string and id are strings, and cgpa is a float value.	*
2	Declare a structure of students with three member variables (name, id and cgpa), where name is a string and id are strings, and cgpa is a float value with **default value**s.	*
3	Given a structure **student**, which has three member variables (name, id and cgpa), declare a variable of structure **student**.	*
4	Given a structure **student**, which has three member variables (name, id and cgpa), declare a variable of structure **student**. Display the value of the member variables.	*
5	Given a structure **student**, which has three member variables (name, id and cgpa), declare a variable of structure **student**. Assign values to the member variables.	*
6	Given a structure **student**, which has three member variables (name, id and cgpa), declare a variable of structure **student**. Populate the member variables from the keyboard.	*

<table>
<tr>
<td>7</td>
<td>Declare a structure of students with three variables (name, id and cgpa). Take information of two students as input and show the output.
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>Shakib Al Hasan
101
3.5
Tamim Iqbal
102
2.7</td><td>Shakib Al Hasan
101
3.5
Tamim Iqbal
102
2.7</td></tr>
</table>
</td>
<td>*</td>
</tr>
<tr>
<td>8</td>
<td>Declare a structure of students with three variables (name, id and cgpa). Now take the input of two students and print the information of that student who has the higher cgpa.
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>Shakib Al Hasan
101
3.5
Tamim Iqbal
102
2.7</td><td>Shakib Al Hasan
101
3.5</td></tr>
</table>
</td>
<td>*</td>
</tr>
</table>

<table>
<tr>
<td>9</td>
<td>Declare a structure of students with three variables (name, id and cgpa). Now take the input of two students and print the information of that student who has better cgpa with a function.
<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td>Shakib Al Hasan
101
3.5
Tamim Iqbal
102
2.7</td><td>Shakib Al Hasan
101
3.5</td></tr>
</table>
</td>
<td>**</td>
</tr>
<tr>
<td>10</td>
<td>You have to declare a structure named triangle. triangle_id, base and height are the members of this structure. Now you will have to take input of three triangles and find out the area of each triangle.
[Triangle Area = (base*height)/2]
<table>
<tr><td>Sample Input
1
5
8
2
4
6
3
3
4</td><td>Sample Output
Area of 1 = 20
Area of 2 = 12
Area of 3 = 6</td></tr>
</table>
</td>
<td>*</td>
</tr>
</table>

<table>
<tr>
<td>11</td>
<td>You have to declare a structure named triangle. triangle_id, base and height are the members of this structure. Now you will have to take input of three triangles and find out which triangle has the maximum area using a function.

[Triangle Area = (base*height)/2]

<table>
<tr><td>Sample Input</td><td>Sample Output</td></tr>
<tr><td>1
5
8
2
4
6
3
3
4</td><td>Area of 1 = 20</td></tr>
</table>
</td>
<td>**</td>
</tr>
</table>

12

The Tigers have clinched a stunning victory over their rivals recently. In that series of three matches, some players put up some amazing performances. Now you have to create a structure named player where you have to store the following information of each player:

1. Player’s name

2. Player’s country

3. Array(size 3) to store runs of 3 matches

4. Array(size 3) to store wickets of 3 matches

5. Array(size 3) to store points of 3 matches

Count points using the following formula:

1. Each wicket = 12 points

2. Runs <=25 in a match = 5 points

3. 25< Runs<=50 in a match = 10 points

4. 50< Runs<=75 in a match = 15 points

5. 75< Runs in a match = 20 points

Now, take input of two players and calculate the points for each player for all the three matches.

Sample Input	Sample Output

**

Shakib Al Hasan Bangladesh 20 75 103 1 1 5 Tamim Iqbal Bangladesh 100 109 17 0 0 0	Match 1: Shakib Al Hasan points: 17 Tamim Iqbal points: 20 Match 2: Shakib Al Hasan points: 27 Tamim Iqbal points: 20 Match 3: Shakib Al Hasan points: 80 Tamim Iqbal points: 5

13

The Tigers have clinched a stunning victory over their rivals recently. In that series of three matches, some players put up some amazing performances. Now you have to create a structure named player where you have to store the following information of each player:

1. Player’s name

2. Player’s country

3. Array(size 3) to store runs of 3 matches

4. Array(size 3) to store wickets of 3 matches

5. Array(size 3) to store points of 3 matches

Count points using the following formula:

1. Each wicket = 12 points

2. Runs <=25 in a match = 5 points

3. 25< Runs<=50 in a match = 10 points

4. 50< Runs<=75 in a match = 15 points

5. 75< Runs in a match = 20 points

Now, take input of two players and calculate the points for each player for all the three matches. And also find man of the match(MOM) for each match based on their points and find out the man of the series on more points overall.

Sample Input	Sample Output

***

Shakib Al Hasan Bangladesh 20 75 103 1 1 5 Tamim Iqbal Bangladesh 100 109 17 0 0 0	Match 1: Shakib Al Hasan points: 17 Tamim Iqbal points: 20 MOM : Tamim Iqbal Match 2: Shakib Al Hasan points: 27 Tamim Iqbal points: 20 MOM : Shakib Al Hasan Match 3: Shakib Al Hasan points: 80 Tamim Iqbal points: 5 MOM : Shakib Al Hasan Man of the Series: Shakib Al Hasan

14

You have been assigned the task of developing a program to manage the inventory of a bookstore. Each book in the inventory has specific details including its title, author, ISBN, and quantity in stock. The program should allow users to add books, search for books by title, update stock quantities, and display the current inventory.

Requirements:

Book Details: Each book has the following details:

- Title (a string)
- Author (a string)
- ISBN (a string)
- Quantity in stock (an integer)

Create a structure named Book to store book details.

Functions:

- void addBook(Book *inventory, int *numBooks): This function enables the bookstore management to add a new book to the inventory. Parameters include:
  inventory: A pointer to an array of Book structures.
  numBooks: A pointer to an integer representing the current number of books in the inventory.
- void searchBook(Book *inventory, int numBooks, char *searchTitle): This function allows users to search for books by title. It should display details of books that match the search title.
- void updateStock(Book *inventory, int numBooks, char *searchTitle, int newQuantity): This function enables users to update the stock quantity of a book with a given title.
- void displayInventory(Book *inventory, int numBooks): This function displays the details of all books currently in the inventory.
- In the main() function:
  Create an array of Book structures to represent the bookstore's inventory.
  Offer a menu that allows users to:
  Add books, Search for books, Update stock quantities, Display the inventory, Exit the program

Constraints:

The bookstore can manage a maximum of 100 books. The title, author, and ISBN strings can be a maximum of 255 characters long.

Sample I/O

```
Menu:
1. Add Book
2. Search Book
3. Update Stock
4. Display Inventory
5. Exit
Enter your choice: 1

Enter book details:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
```

***

```
ISBN: 978-0743273565
Quantity in Stock: 15

Book added successfully.

Menu:
1. Add Book
2. Search Book
3. Update Stock
4. Display Inventory
5. Exit
Enter your choice: 2

Enter the title to search: The Great Gatsby

Book Found:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
ISBN: 978-0743273565
Quantity in Stock: 15

Menu:
1. Add Book
2. Search Book
3. Update Stock
4. Display Inventory
5. Exit
Enter your choice: 3

Enter the title to update stock: The Great Gatsby
Enter the new quantity: 10

Stock quantity updated successfully.

Menu:
1. Add Book
2. Search Book
3. Update Stock
4. Display Inventory
5. Exit
Enter your choice: 4

Book Inventory:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
ISBN: 978-0743273565
Quantity in Stock: 10
```

```
Menu:
1. Add Book
2. Search Book
3. Update Stock
4. Display Inventory
5. Exit
Enter your choice: 5

Exiting program.
```

15

You are developing a program to manage student records for a school. Each student has specific details including their name, roll number, and marks in different subjects. The program should allow users to add students, search for students by roll number, calculate average marks, and display student details.

Student Details: Each student has the following details:

- Name (a string)
- Roll number (an integer)
- Marks in different subjects (an array of integers)

Task 1: Create a structure named Student to store student details.

Functions:

- void addStudent(Student *record, int *numStudents): This function allows school administration to add a new student's details to the record. Parameters include:
  record: A pointer to an array of Student structures.
  numStudents: A pointer to an integer representing the current number of students in the record.
- void searchStudent(Student *record, int numStudents, int searchRollNumber): This function allows users to search for students by roll number. It should display details of students that match the search roll number.
- float calculateAverageMarks(Student *record, int numStudents, int searchRollNumber): This function calculates and returns the average marks of a student with a given roll number.
- void displayStudentDetails(Student *record, int numStudents, int searchRollNumber): This function displays the details of a student with a specified roll number, including their average marks.
- In the main() function:
  Create an array of Student structures to represent the school's student records.
  Provide a menu that allows users to:
  Add students, Search for students, Calculate average marks, Display student details, Exit the program

Constraints:

The school can manage a maximum of 100 students.

The name strings can be a maximum of 255 characters long.

Each student has marks in 5 subjects.

***

Sample I/O

```
Menu:
1. Add Student
2. Search Student
3. Calculate Average Marks
4. Display Student Details
5. Exit
Enter your choice: 1

Enter student details:
Name: Alice Johnson
Roll Number: 101
Marks in 5 subjects: 85 90 78 92 88

Student added successfully.

Menu:
1. Add Student
2. Search Student
3. Calculate Average Marks
4. Display Student Details
5. Exit
Enter your choice: 2

Enter the roll number to search: 101

Student Found:
Name: Alice Johnson
Roll Number: 101
Marks in 5 subjects: 85 90 78 92 88

Menu:
1. Add Student
2. Search Student
3. Calculate Average Marks
4. Display Student Details
5. Exit
Enter your choice: 3

Enter the roll number to calculate average marks: 101

Average Marks: 86.60

Menu:
1. Add Student
2. Search Student
3. Calculate Average Marks
```

```
4. Display Student Details
5. Exit
Enter your choice: 4

Enter the roll number to display details: 101

Student Details:
Name: Alice Johnson
Roll Number: 101
Marks in 5 subjects: 85 90 78 92 88
Average Marks: 86.60

Menu:
1. Add Student
2. Search Student
3. Calculate Average Marks
4. Display Student Details
5. Exit
Enter your choice: 5

Exiting program.
```