**1) Question: What is the default value assigned to array elements in C#?**

In C#, the default values for array elements are 0 for numeric types, false for bool, null for reference types, and default field values for structs. These defaults are automatically assigned upon array initialization.

**2) Question: What is the difference between Array.Clone() and Array.Copy()?**

Array.Clone() creates a shallow copy of the array, duplicating the array structure but not the objects it references. Array.Copy() copies elements from one array to another, allowing you to specify the target array and range of elements to copy.

**3) Question: What is the difference between GetLength() and Length for multi-dimensional arrays?**

For multidimensional arrays, GetLength(dimension) returns the size of a specific dimension, while Length returns the total number of elements across all dimensions.

**4) Question: What is the difference between Array.Copy() and Array.ConstrainedCopy()?**

Array.Copy() is a general-purpose method for copying elements between arrays, without runtime checks. Array.ConstrainedCopy() performs additional runtime checks to ensure the operation doesn't partially succeed, throwing an exception if it fails.

**5) Question: Why is foreach preferred for read-only operations on arrays?**

foreach is preferred for read-only operations on arrays because it automatically handles iteration and avoids accidental modification of elements. It provides cleaner, more concise code when the array elements are not intended to be changed.

**6) Question: Why is input validation important when working with user inputs?**

Input validation is important to ensure that user inputs are accurate, secure, and in the expected format and preventing errors.

**7) Question: How can you format the output of a 2D array for better readability?**

To format the output of a 2D array for better readability, you can use nested loops to print each row on a new line and separate elements with spaces or tabs.

**8) Question: When should you prefer a switch statement over if-else?**

A switch statement is preferred over if-else when you need to compare a single variable against multiple values, as it is more concise and often more efficient. It improves readability and performance in cases with many conditions.

**9) Question: What is the time complexity of Array.Sort()?**

The time complexity of Array.Sort() is O(n log n) in the average and worst cases, This is typically achieved using the Quicksort or Heapsort algorithms

**10) Question: Which loop (for or foreach) is more efficient for calculating the sum of an array, and why?**

The for loop is generally more efficient for calculating the sum of an array because it provides direct access to the array indices, avoiding the overhead of enumerating elements. foreach involves an iterator, which can be slightly slower in comparison.

**3- What happens if the user enters a value outside the range of 1 to 7?**

If the user enters a value outside the range of 1 to 7, the Enum.Parse() method will throw an ArgumentException. The catch block will handle this exception, and the program will output an error message: "Invalid input. Please enter a number between 1 and 7."