1- What is the purpose of the finally block?
The finally block is always executed after the try and catch blocks, regardless of whether an exception occurred. It is typically used to release resources, close files, or perform cleanup operations.

2- How does int.TryParse() improve program robustness compared to int.Parse()?
int.TryParse() prevents exceptions from being thrown if the input cannot be converted to an integer. Instead, it returns false, allowing the program to handle invalid input gracefully without crashing.

3- What exception occurs when trying to access Value on a null Nullable<T>?
Accessing .Value on a null Nullable<T> throws an InvalidOperationException because there is no value to retrieve.

4- Why is it necessary to check array bounds before accessing elements?
Accessing an index outside the valid range causes an IndexOutOfRangeException. Checking bounds prevents runtime errors and ensures safe access to array elements.

5- How is the GetLength(dimension) method used in multi-dimensional arrays?
GetLength(dimension) returns the number of elements along a specified dimension (0 for rows, 1 for columns, etc.) in a multi-dimensional array.

6- How does the memory allocation differ between jagged arrays and rectangular arrays?

- **Jagged arrays**: Each row is a separate array stored in different memory locations; rows can have different lengths.
- **Rectangular arrays**: All elements are stored in a single contiguous memory block with fixed row and column sizes.

7- What is the purpose of nullable reference types in C#?
Nullable reference types help prevent NullReferenceException by allowing developers to explicitly declare when a reference type can be null, enabling compiler warnings for potential null usage.

8- What is the performance impact of boxing and unboxing in C#?
Boxing and unboxing cause extra memory allocation on the heap and require type conversion, which impacts performance, especially in performance-critical or high-frequency operations.

9- Why must out parameters be initialized inside the method?
out parameters must be assigned before the method returns because the caller expects them to contain a value after the method finishes execution.

10- **Why must optional parameters always appear at the end of a method's parameter list?**
Optional parameters must be at the end so that the compiler can correctly match arguments to parameters without ambiguity.

11- How does the null propagation operator prevent NullReferenceException?
The ?. operator checks if the object is null before accessing its members. If the object is null, it returns null instead of throwing a NullReferenceException.

12- When is a switch expression preferred over a traditional if statement?
Switch expressions are preferred when you need to match a single value against multiple patterns in a clean, concise, and more readable way compared to multiple if-else statements.

13- What are the limitations of the params keyword in method definitions?

- Only one params parameter is allowed per method.
- It must be the last parameter in the method signature.
- All arguments must be of the same type as the params array element type.