1. What is the purpose of the finally block?
   Used to ensure that a piece of code always executes, whether or not an exception occurs in the try block.

2. How does int.TryParse() improve program robustness compared to int.Parse()?
   Cause it doesn't throw exception but return false if the input was invalid, and no need to use try catch blocks.

3. What exception occurs when trying to access Value on a null Nullable<T>?
   InvalidOperationException is thrown

4. Why is it necessary to check array bounds before accessing elements?
   To avoid index out of range exception.

5. How is the GetLength(dimension) method used in multi-dimensional arrays?
   Used to return the number of elements in a specific dimension of the array.

6. How does the memory allocation differ between jagged arrays and rectangular arrays?
   - Rectangular arrays Stored in one single contiguous block in memory.
   - Jagged arrays: Outer array stores references to inner arrays, each inner array is stored separately in memory

7. What is the purpose of nullable reference types in C#?
   The main goal is to make null handling explicit in your code by:
   - Warning you when you might be dereferencing a null reference.
   - Forcing you to declare when a reference can be null and when it cannot.

8. What is the performance impact of boxing and unboxing in C#?
   Boxing: Allocates a new object on the heap, Copies the value type's data into that object.

   Unboxing: Checks that the object is indeed a boxed value of the correct type, Copies the data back to a value type variable.

9. Why must out parameters be initialized inside the method?
   Cause it made to return value so it must be assigned inside the method before calling.

10. Why must optional parameters always appear at the end of a method's parameter list?
    If it was in any other position you won't be able to skip it without naming all parameters, and if you don't named it the compiler won't know witch parameter has this value.

11. How does the null propagation operator prevent NullReferenceException?
    By using it you tell the compiler that this variable may be null.

12. When is a switch expression preferred over a traditional if statement?
    When the condition is concerned with only one value.

13. What are the limitations of the params keyword in method definitions?
    You can use it only once in a method and all the list must be the same type.