

1. What is the difference between `int.Parse` and `Convert.ToInt32` when handling null inputs?
 - `Parse`: throw `ArgumentNullException`.
 - `Convert`: doesn't throw exception just return 0.
2. Why is `TryParse` recommended over `Parse` in user-facing applications?
 - `Parse` will throw exception if the input is not valid, but `TryParse` will return `bool`, `true` if input is valid and converted successfully, otherwise will return `false`.
3. Explain the real purpose of the `GetHashCode()` method.
 - To return a numeric value that helps quickly identify objects, especially when used in hash-based collections
4. What is the significance of reference equality in .NET?
 - Reference equality: Two object variables point to the same memory location, they reference the exact same object instance.
 - Reference equality is about whether two variables point to the same object, not whether their contents are the same.
5. Why string is immutable in C#?
 - Security: Prevents tampering with sensitive values.
 - Hashing: Keeps Dictionary and HashSet keys stable
 - Thread-Safety: Strings are safe to share between threads
 - Interning: Saves memory by reusing strings
 - Predictability: Easier debugging and logic tracking
6. How does `StringBuilder` address the inefficiencies of string concatenation?
 - In normal string when concatenation it creates a new object in the memory but with `StringBuilder` it modifies on the same object.
7. Why is `StringBuilder` faster for large-scale string modifications?
 - If we used strings there would be performance issues within the memory but string builder keep change on the same object till it's block become full then move to another block but string move to another with every single change.
8. Which string formatting method is most used and why?
 - The most used is the Interpolation method
 - It make the code cleaner and easy to understand, avoid place holder mismatch, variables can go directly inside the string so it more readable.
9. Explain how `StringBuilder` is designed to handle frequent modifications compared to strings.
 - Normal String on every modification it creates a new object.
 - `StringBuilder` modifies the same object.

Part02

1. What's Enum data type, when is it used? And name three common built-in Enums used frequently?
 - Enum is a value type in C# that lets you define a set of named constants.
 - Use Enum when:
 - You have a limited, fixed set of related options
 - You want to make code self-documenting
 - You need to group integer-based options under readable names
 - Built-in Enums (DayOfWeek, ConsoleColor, DateTimeKind).

2. What are scenarios to use string Vs StringBuilder?
 - Use string In: a simple code with few strings, you don't need to make a lot modifications, performance is not issue for you.
 - StringBuilder in large scale project with a lot strings that you need to modify frequently, when using more temp strings in the memory, when you care about performance.