Task Day03

Question: What is the difference between int.Parse and Convert.ToInt32 when handling null inputs?

The difference between int.Parse and Convert.ToInt32 when handling null inputs is:

> -int.Parse(null) throws an ArgumentNullException because it does not accept null values.
> -Convert.ToInt32(null) does not throw an exception; instead, it returns 0.

However, both methods throw a FormatException if the input string is not a valid numeric value.

Question: Why is TryParse recommended over Parse in user-facing applications?

TryParse is recommended over Parse in user-facing applications because it prevents runtime exceptions when the input is invalid.

> **Parse()** throws an exception if the input is null or not in a valid numeric format.
>
> **TryParse()** does not throw an exception. Instead, it returns a boolean value:
> -true if the conversion succeeds
> -false if the conversion fails

This makes TryParse safer and more efficient when handling user input, since users may enter invalid data.

Question: Explain the real purpose of the GetHashCode() method

The real purpose of **GetHashCode()** is to generate a numeric hash value used to efficiently store and retrieve objects in hash-based collections like Dictionary and HashSet.

Question: What is the significance of reference equality in .NET?

Reference equality in .NET checks whether two references point to the same object in memory, not whether their values are equal.

Question: Why string is immutable in C# ?

String is immutable in C# because String is an array of chars, so it is fixed size so it is immutable.

Question: How does StringBuilder address the inefficiencies of string concatenation?

StringBuilder improves performance by modifying text in place without creating a new object for every concatenation, making it more efficient for repeated string operations.

Question: Why is StringBuilder faster for large-scale string modifications?

StringBuilder is faster for large-scale string modifications because it modifies text in place without creating new objects for each change.

Question: Which string formatting method is most used and why?

String Interpolation ($) is the most used string formatting method because it is clear, readable, less error-prone, and allows inline expressions.

Question: Explain how StringBuilder is designed to handle frequent modifications compared to strings.

StringBuilder is designed for frequent modifications by using a mutable internal buffer that allows text changes in place, avoiding the creation of new objects like with immutable strings, which improves performance and memory usage.

What's Enum data type, when is it used? And name three common built_in enums used frequently?

Enum is a special data type that defines a set of named constants. It is used to represent fixed sets of related values for readability, safety, and maintainability. Common built-in enums: **ConsoleColor, DayOfWeek, FileMode.**

What are scenarios to use string Vs StringBuilder?

Use string for short, rarely modified text. Use StringBuilder for large-scale or frequently modified text to improve performance and reduce memory usage.