**Question#01:**

Describe functioning of regex C# library, give examples of patterns, seperators and anchors etc.

**Answer:**

Regular expressions (regex) in C# are implemented through the System.Text.RegularExpressions namespace. This library provides classes and methods for working with regular expressions to search, match, and manipulate strings based on specific patterns. Here's a brief overview of the key elements and functions of the C# regex library, along with some examples:

1. **Regex Class:**

The Regex class is the primary class used for working with regular expressions in C#. It provides methods for matching patterns in strings.

**using System.Text.RegularExpressions;**

1. **Patterns:**

Patterns are the core of regular expressions. They consist of a combination of characters and Meta characters that define the search criteria. Here are some common patterns and Meta characters:

.: Matches any character except a newline.

\*: Matches the preceding element zero or more times.

+: Matches the preceding element one or more times.

?: Matches the preceding element zero or one time.

[]: Matches any single character listed inside the square brackets.

[^]: Matches any single character not listed inside the square brackets.

|: Acts as an OR operator.

(): Groups patterns together.

1. **Anchors:**

Anchors are used to match patterns at specific positions within a string.

^: Matches the start of a string.

$: Matches the end of a string.

\b: Matches a word boundary.

\B: Matches a non-word boundary.

**Modifiers:**

Modifiers affect how the pattern is matched.

i: Case-insensitive matching.

m: Multi-line matching (e.g., ^ and $ match the start/end of each line).

s: Single-line matching (dot . matches newline characters as well).

Now, let's look at some examples of using regex in C#:

**Example 1:**

**Simple Pattern Matching**

string input = "Hello, World! This is a simple example.";

string pattern = "World";

Match match = Regex.Match(input, pattern);

if (match.Success)

{

Console.WriteLine($"Match found at index {match.Index}: {match.Value}");

}

**Example 2:**

**Pattern with Anchors and Modifiers**

string input = "The quick brown fox\nJumps over the lazy dog";

string pattern = "^The|^Jumps";

Regex regex = new Regex(pattern, RegexOptions.Multiline);

MatchCollection matches = regex.Matches(input);

foreach (Match match in matches)

{

Console.WriteLine($"Match: {match.Value}");

}

**Example 3:**

**Character Classes and Quantifiers**

string input = "Emails: john@example.com, alice@test.org, and bob@gmail.com";

string pattern = @"\b\w+@\w+\.\w+\b";

MatchCollection matches = Regex.Matches(input, pattern);

foreach (Match match in matches)

{

Console.WriteLine($"Email: {match.Value}");

}

**Example 4:**

**Substitution**

string input = "Hello, my name is John. Hi, I'm Alice.";

string pattern = "John|Alice";

string replacement = "Jane";

string result = Regex.Replace(input, pattern, replacement);

Console.WriteLine(result);