Muhammad Abdullah Arshad (SP20-BCS-033)

**Question 1**

**Describe functioning of regex C# library , give examples of patterns, seperators and anchors e.t.c.**

**Answer:** Regular expressions, often abbreviated as "regex," are a powerful tool for pattern matching and text manipulation. In C#, you can work with regular expressions using the System.Text.RegularExpressions namespace, which provides a Regex class. This class allows you to create, compile, and work with regular expressions in your C# code.

Here's a breakdown of how the regex library in C# functions, along with examples of patterns, separators, and anchors:

**1. Creating a Regex Object:**

To work with regular expressions in C#, you first create a Regex object by providing a pattern as a string. For example:

string pattern = @"\b\d{3}-\d{2}-\d{4}\b"; // Pattern for matching social security numbers

Regex regex = new Regex(pattern);

**2. Pattern Elements:**

Literals: Characters in the pattern that match themselves. For example, in the pattern "cat," the literal "cat" will match the string "cat."

Metacharacters: Special characters with predefined meanings. Examples include ., \*, +, ?, |, (), [], {}, \, and more.

**3. Quantifiers:**

\*: Matches zero or more occurrences of the preceding element.

+: Matches one or more occurrences of the preceding element.

?: Matches zero or one occurrence of the preceding element.

{n}: Matches exactly n occurrences of the preceding element.

{n,}: Matches at least n occurrences of the preceding element.

{n,m}: Matches between n and m occurrences of the preceding element.

**4. Character Classes:**

[abc]: Matches any single character 'a', 'b', or 'c.'

[^abc]: Matches any single character except 'a', 'b', or 'c.'

[0-9]: Matches any single digit.

**5. Anchors:**

^: Matches the start of a line.

$: Matches the end of a line.

\b: Matches a word boundary.

**6. Modifiers:**

i: Case-insensitive matching.

m: Multi-line mode, where ^ and $ match the start/end of each line.

**7. Special Sequences:**

\d: Matches a digit (equivalent to [0-9]).

\w: Matches a word character (equivalent to [a-zA-Z0-9\_]).

\s: Matches a whitespace character (e.g., space, tab, newline).

**8. Grouping and Capturing:**

(abc): Creates a capturing group for the enclosed pattern.

(?<name>abc): Creates a named capturing group.

Here are some examples of using these elements:

// Example 1: Match a date in the format "dd/mm/yyyy"

string datePattern = @"\b\d{2}/\d{2}/\d{4}\b";

Regex dateRegex = new Regex(datePattern);

// Example 2: Match email addresses

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

Regex emailRegex = new Regex(emailPattern);

// Example 3: Match phone numbers with optional area code

string phonePattern = @"(\d{3}-)?\d{3}-\d{4}";

Regex phoneRegex = new Regex(phonePattern);