# Registration fa20-bcs-063

# Name Tanveer Ahmed

# Answer no 1

In C#, you can use regular expressions (regex) through the `System.Text.RegularExpressions` namespace. Regular expressions are patterns used to match and manipulate strings. Here's an overview of how the regex library in C# works, along with some common regex elements:

## 1. \*Regex Pattern Creation\*:

You create a regex pattern using a string, which contains a sequence of characters and special symbols that define the pattern you want to match.

## 2. \*Regex Options\*:

You can specify regex options to control how the pattern matching works. Common options include `RegexOptions.IgnoreCase`, `RegexOptions.Multiline`, and `RegexOptions.Singleline`.

## 3. \*Matching\*:

You use the `Regex.Match` method to search for a pattern within a given input string and return the first match.

## 4. \*Matching All Occurrences\*:

To find all occurrences of a pattern in a string, you can use `Regex.Matches`.

## 5. \*Pattern Anchors\*:

- `^` - Matches the start of a line.

- `$` - Matches the end of a line.

## 6. \*Character Classes\*:

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

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

- `[a-z]` - Matches any lowercase letter from 'a' to 'z'.

## 7. \*Quantifiers\*:

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

- `+` - Matches one or more occurrences.

- `?` - Matches zero or one occurrence.

- `{n}` - Matches exactly 'n' occurrences.

- `{n, m}` - Matches between 'n' and 'm' occurrences.

## 8. \*Escape Sequences\*:

You can use backslashes to escape special characters. For example, `\.` matches a period.

## 9. \*Word Boundaries\*:

- `\b` - Matches a word boundary (start or end of a word).

- `\B` - Matches a non-word boundary.

## 10. \*Groups and Capturing\*:

You can use parentheses `()` to create groups, which allow you to capture portions of the matched text.

## 11. \*Alternation\*:

- `|` - Acts as an OR operator. For example, `A|B` matches 'A' or 'B'.

## 12. \*Common Separators\*:

- `\s` - Matches any whitespace character.

- `\S` - Matches any non-whitespace character.

- `\d` - Matches any digit.

- `\D` - Matches any non-digit.

- `\w` - Matches any word character (alphanumeric or underscore).

- `\W` - Matches any non-word character.

### Examples:

csharp

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

string pattern = @"\b\w{5}\b"; // Matches 5-letter words

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

foreach (Match match in matches)

{

Console.WriteLine(match.Value);

}

This code will find and print all 5-letter words in the input string.

## Regex

Regex is a powerful tool for string manipulation and pattern matching in C#. The above examples demonstrate some basic patterns, but regex can be much more complex and versatile depending on your needs.

# Answer no 2

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ccmid

{

internal class LL1Parser

{

private string[] tokens;

private int index;

public LL1Parser(string input)

{

// Tokenize the input string (split by comma)

tokens = input.Split(',');

index = 0;

}

public void Parse()

{

if (List())

{

if (index == tokens.Length)

{

Console.WriteLine("Parsing successful!");

}

else

{

Console.WriteLine("Parsing error. Unexpected tokens.");

}

}

else

{

Console.WriteLine("Parsing error.");

}

}

private bool List()

{

if (Item())

{

return Rest();

}

return false;

}

private bool Rest()

{

if (index < tokens.Length && tokens[index] == ",")

{

index++;

if (Item())

{

return Rest();

}

return false;

}

return true; // ε (empty string)

}

private bool Item()

{

if (index < tokens.Length && (tokens[index] == "id" || tokens[index] == "num" || tokens[index] == "string"))

{

index++;

return true;

}

return false;

}

}

}

…………………………………………..

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ccmid

{

internal class Program

{

static void Main(string[] args)

{

// Input string to parse

string input = "id, num, string, id, num";

// Create the parser

LL1Parser parser = new LL1Parser(input);

// Start parsing

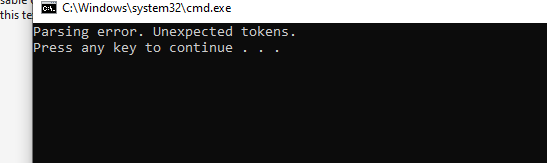
parser.Parse();

}

}

}

# Output:



# Answer no 3

Code:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace passgen

{

internal class Program

{

public static string GeneratePassword(string firstName, string lastName, string registrationNumber)

{

// Initials of the first and last name in uppercase

string initials = firstName[0].ToString().ToUpper() + lastName[0].ToString().ToUpper();

// Last two digits of the registration number

string lastTwoDigits = registrationNumber.Substring(registrationNumber.Length - 2);

// Generate random special characters

string specialCharacters = GenerateRandomSpecialCharacters(2);

// Generate random numbers

string numbers = GenerateRandomNumbers(4);

// Combine all elements and shuffle the password

string password = initials + lastTwoDigits + specialCharacters + numbers;

password = new string(password.ToCharArray().OrderBy(x => Guid.NewGuid()).ToArray());

// Ensure the password length does not exceed 20 characters

if (password.Length > 20)

{

password = password.Substring(0, 20);

}

return password;

}

private static string GenerateRandomSpecialCharacters(int count)

{

string specialChars = "!@#$%^&\*()\_-+=<>?";

Random random = new Random();

StringBuilder result = new StringBuilder();

for (int i = 0; i < count; i++)

{

int index = random.Next(specialChars.Length);

result.Append(specialChars[index]);

}

return result.ToString();

}

private static string GenerateRandomNumbers(int count)

{

string numbers = "0123456789";

Random random = new Random();

StringBuilder result = new StringBuilder();

for (int i = 0; i < count; i++)

{

int index = random.Next(numbers.Length);

result.Append(numbers[index]);

}

return result.ToString();

}

public static void Main(string[] args)

{

string firstName = "Tanveer";

string lastName = "Ahmed";

string registrationNumber = "fa20-bcs-063";

string password = GeneratePassword(firstName, lastName, registrationNumber);

Console.WriteLine("Generated Password: " + password);

}

}

}

# Output:

