Step-by-Step Explanation of the C# Program

Overview

This C# program performs various string operations based on user inputs. It:

1. Checks if a substring exists within a given string.
2. Replaces a specific character in the string.
3. Swaps the case of each character in the string.
4. Removes whitespace from the string.
5. Counts the occurrences of each letter in the string.

Step 1: Taking User Input

The program takes four inputs from the user:

- **Main String** (full string to process)
- **Sub String** (to check existence in the main string)
- **Character to be replaced**
- **Replacement character**

The `GetInput` method is used to read inputs from the console.

```
static string GetInput(string prompt)
{
    Console.WriteLine(prompt);
    return Console.ReadLine();
}
```

This method prints a prompt (empty in this case) and reads user input.

Step 2: Checking Substring Existence

The `CheckSubstringExists` method checks if the substring exists in the main string using the `Contains` method.

```
static bool CheckSubstringExists(string main, string sub)
{
    return main.Contains(sub);
}
```

Example:

```
Main String: HelloWorld
Sub String: World
Output: Yes (Substring exists)
```

Step 3: Replacing a Character

The `ReplaceCharacter` method replaces occurrences of `oldChar` with `newChar` in the main string.

```
static string ReplaceCharacter(string input, char oldChar, char
newChar)
{
    return input.Replace(oldChar, newChar);
}
```

Example:

```
Main String: HelloWorld
Character to replace: o
Replacement character: x
Output: HellxWxrld
```

Step 4: Swapping Case

The `SwapCase` method converts uppercase letters to lowercase and vice versa.

```
static string SwapCase(string input)
{
    return new string(input.Select(c => char.IsLetter(c) ?
(char.IsUpper(c) ? char.ToLower(c) : char.ToUpper(c)) :
c).ToArray());
}
```

Example:

```
Main String: HelloWorld
Output: hELLOwORLD
```

Step 5: Removing Whitespace

The `RemoveWhitespace` method removes all spaces from the string.

```
static string RemoveWhitespace(string input)
{
    return string.Concat(input.Where(c => !char.IsWhiteSpace(c)));
}
```

Example:

```
Main String: Hello World
Output: HelloWorld
```

Step 6: Counting Letter Occurrences

The `CountLetters` method counts occurrences of each letter in the main string and returns a dictionary.

```
static Dictionary<char, int> CountLetters(string input)
{
    return input.GroupBy(c => c).Where(g =>
char.IsLetter(g.Key)).ToDictionary(g => g.Key, g => g.Count());
}
```

Example:

```
Main String: HelloWorld
Output: H:1, e:1, l:3, o:2, W:1, r:1, d:1
```

Test Cases

Test Case	Main String	Substring	Char to Replace	Replacement Char	Expected Output
1	HelloWorld	World	o	x	Yes, HellxWxrld, hELLOwORLD, HelloWorld, H:1, e:1, l:3, o:2, W:1, r:1, d:1
2	CSharpCode	Sharp	C	Z	Yes, ZSharpZode, cSHARPcODE, CSharpCode, C:2, S:1, h:1, a:1, r:1, p:1, o:1, d:1, e:1
3	Test Case	Case	e	i	Yes, Tist Casi, tEST cASE, TestCase, T:1, e:2, s:1, t:1, C:1, a:1

Test Case #	Input String	Operation	Expected Output	Remarks
1	`"Welcome to Programming!"`	Reverse String	`"!gnimmargorP ot emocleW"`	Basic case
2	`"C# is fun"`	Convert to Uppercase	`"C# IS FUN"`	Checks case conversion
3	`"DATA Science"`	Convert to Lowercase	`"data science"`	Checks case conversion

Test Case #	Input String	Operation	Expected Output	Remarks
4	`" Trim this! "`	Trim Whitespace	`"Trim this!"`	Leading/trailing spaces
5	`"abc,def,ghi"`	Split by `,`	`["abc", "def", "ghi"]`	Splitting test
6	`"Find the word"`	Search `"word"`	`True`	Word exists
7	`"Find the text"`	Search `"missing"`	`False`	Word does not exist
8	`"Replace me"`	Replace `"me"` with `"you"`	`"Replace you"`	Substring replacement
9	`""`	Reverse String	`""`	Edge case: empty string
10	`"12345"`	Reverse String	`"54321"`	Numeric string reversal