

## Aufgabe 1: Längen-Validierung

- **Ziel:** Prüfen, ob eine Eingabe eine erwartete Länge hat.
- **Anforderungen:**
  1. Bitten Sie den Benutzer um die Eingabe einer 5-stelligen Postleitzahl.
  2. Prüfen Sie mit der `Length`-Eigenschaft, ob die Eingabe genau 5 Zeichen lang ist.
  3. Geben Sie eine entsprechende Meldung aus ("Gültige Länge" oder "Ungültige Länge").

## Aufgabe 2: Vokale zählen

- **Ziel:** Die Anzahl der Vokale (a, e, i, o, u) in einem Text zählen.
- **Anforderungen:**
  1. Lesen Sie einen Text vom Benutzer ein.
  2. Erstellen Sie eine Zählvariable und initialisieren Sie sie mit 0.
  3. Iterieren Sie mit einer `foreach`-Schleife über jedes Zeichen.
  4. Prüfen Sie in der Schleife, ob das aktuelle Zeichen ein Vokal ist (Groß- und Kleinschreibung ignorieren).
  5. Erhöhen Sie den Zähler, falls es ein Vokal ist, und geben Sie am Ende die Gesamtanzahl aus.

## Aufgabe 3: Palindrom-Prüfung

- **Ziel:** Prüfen, ob ein Wort vorwärts wie rückwärts gelesen dasselbe ist (z.B. "Anna", "Lagerregal").
- **Anforderungen:**
  1. Lesen Sie ein Wort vom Benutzer ein.
  2. Iterieren Sie mit einer `for`-Schleife nur bis zur Mitte des Wortes ( `text.Length / 2` ).
  3. Vergleichen Sie in jeder Iteration das Zeichen am Index `i` mit dem Zeichen am spiegelverkehrten Index `text.Length - 1 - i`.
  4. Wenn ein Paar nicht übereinstimmt, ist es kein Palindrom. Geben Sie eine entsprechende Meldung aus und beenden Sie die Prüfung.
  5. Wenn die Schleife durchläuft, ist es ein Palindrom.

## Aufgabe 4: Text umkehren

- **Ziel:** Einen eingegebenen Text Zeichen für Zeichen umkehren.
- **Anforderungen:**
  1. Lesen Sie einen Text vom Benutzer ein.
  2. Erstellen Sie einen neuen, leeren String für das Ergebnis.
  3. Iterieren Sie mit einer `foreach` - oder `for` -Schleife über den eingegebenen Text.
  4. Fügen Sie in jeder Iteration das aktuelle Zeichen **vor** den bisherigen Ergebnis-String (z.B. `ergebnis = aktuellesZeichen + ergebnis;` ).
  5. Geben Sie den umgekehrten String aus.

## Aufgabe 5: Eingabe normalisieren

- **Ziel:** Eine Benutzereingabe für einen Vergleich vorbereiten.
- **Anforderungen:**
  1. Bitten Sie den Benutzer, "Ja" oder "Nein" einzugeben. Erlauben Sie beliebige Groß-/Kleinschreibung und Leerzeichen (z.B. " jA ").
  2. Nutzen Sie `Trim()` und `ToLower()` an der Eingabe, um sie zu normalisieren.
  3. Vergleichen Sie den normalisierten String mit "ja" oder "nein" und geben Sie eine verständliche Rückmeldung aus.

## Aufgabe 6: Domain extrahieren

- **Ziel:** Aus einer E-Mail-Adresse den Domain-Namen extrahieren.
- **Anforderungen:**
  1. Bitten Sie den Benutzer um die Eingabe einer E-Mail-Adresse.
  2. Finden Sie die Position des `@`-Zeichens mit `IndexOf` .
  3. Nutzen Sie `Substring` , um den Text *nach* dem `@`-Zeichen zu extrahieren.
  4. Geben Sie den extrahierten Domain-Namen aus.

## Aufgabe 7: Text zensieren

- **Ziel:** Ein Wort in einem Satz durch Sternchen ersetzen.
- **Anforderungen:**
  1. Definieren Sie einen Satz, z.B. `"Das ist ein Test, der ein Test ist."` .
  2. Bitten Sie den Benutzer um ein Wort, das zensiert werden soll.
  3. Verwenden Sie `Replace` , um alle Vorkommen des Wortes durch `"****"` zu ersetzen.
  4. Geben Sie das Ergebnis aus.

## Aufgabe 8: Wortzähler

- **Ziel:** Die Anzahl der Wörter in einem Satz zählen.
- **Anforderungen:**
  1. Lesen Sie einen Satz vom Benutzer ein und bereinigen Sie ihn mit `Trim()` .
  2. Initialisieren Sie einen Zähler für die Wörter (z.B. mit 1, wenn der Text nicht leer ist).
  3. Iterieren Sie mit einer Schleife über jedes Zeichen des Satzes.
  4. Erhöhen Sie den Zähler jedes Mal, wenn Sie ein Leerzeichen `' '` finden.
  5. *Zusatz:* Verhindern Sie, dass mehrere Leerzeichen hintereinander den Zähler mehrfach erhöhen.

## Aufgabe 9: Passwortanalyse

- **Ziel:** Eine Konsolenanwendung, die ein Passwort nach einfachen Regeln validiert.
- **Anforderungen:**
  1. Lesen Sie ein Passwort von der Konsole ein.
  2. Prüfen Sie folgende Kriterien:
    - Mindestlänge von 8 Zeichen ( `Length` ).
    - Vorhandensein von mindestens einer Ziffer.
    - Vorhandensein von mindestens einem Großbuchstaben.
  3. Iterieren Sie hierfür über den String und nutzen Sie Hilfsmethoden wie `char.IsDigit()` und `char.IsUpper()` .
  4. Geben Sie eine aussagekräftige Rückmeldung an den Benutzer.