

# Aufgabe 2: Erkennen von Mustern (C)

Implementierungsaufgabe inklusive Anpassung des Unit Testrahmens (siehe Labor "count").

## Erkennen des längsten Suffix nach einem vorgegebenen Muster

Erste Teilaufgabe ist es eine Funktion zu schreiben, die den längsten suffix (Endung) liefert, in der keine zwei hintereinander folgenden Doppelpunkte (:) vorkommen. Als Eingabe betrachte man folgende Strings

```
"Ha::ll::o"  
"47::11"
```

Längste Endungen für obige Beispiele sind

```
"o"  
"11"
```

Verwenden Sie folgenden Funktionsprototypen.

```
char* extract(char* input);
```

In obiger Funktion wird das Ergebnis als return Wert geliefert. Das Ergebnis, ein Zeiger, verweist auf die Stelle im Eingabestring die den längsten suffix (Endung) liefert, in der keine zwei hintereinander folgenden Doppelpunkte (:) vorkommen.

Als Variante, verwenden Sie als Rückgabewert einen Pointer auf einen Pointer. Wieso reicht ein Pointer nicht aus?

```
void extract2(char* input, char** output)
```

## Freiwillige Zusatzaufgabe:

Verallgemeinern Sie Ihre Lösung: Das Muster zur Berechnung des längsten Suffix ist ein zusätzlicher Parameter.

```
char* extract(char* pattern, char* input);
```

# Testrahmen

Die zweite Teilaufgabe ist es den Testrahmen von "Count - Zählen von Wörtern" anzupassen. Eine hinreichende Menge von Testfällen sollte zur Verfügung stehen.

Hinweis: Im Falle von "count" ist das erwartete Ergebnis ein Integerwert. Im Falle von "extract" ist dies ein String. Wie können Sie auf Gleichheit zweier Strings prüfen?

## Source file

Details zu Testrahmen finden sie im Foliensatz "Beispiele".