Zeichenketten

Lernziele

- Sie können eine Zeichenkette als ein Array definieren.
- Sie können Zeichenketten mit Bibliotheksfunktionen bearbeiten.

Zeichenketten

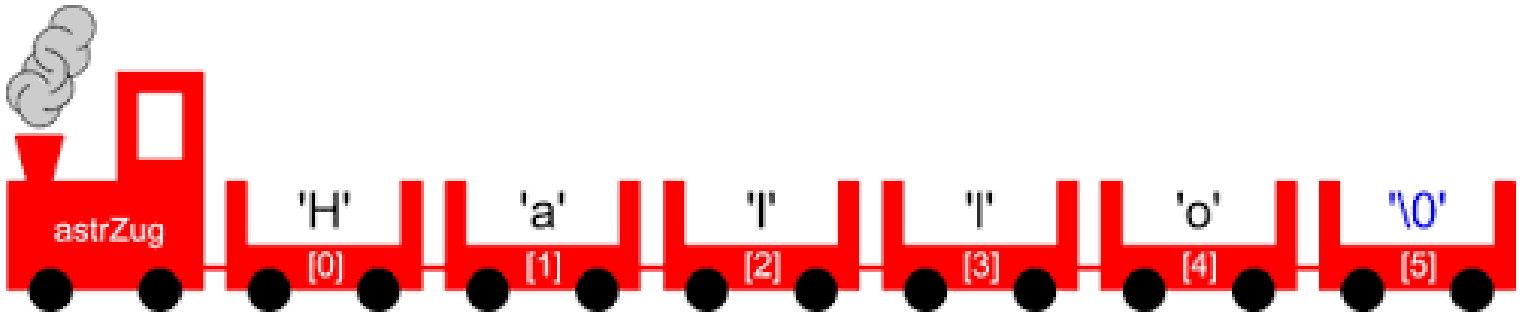
Eine Zeichenkette (String) in C:

- Wird als ein Array vom Typ char definiert.
- Nullterminiert:

  Eine Zeichenkette endet immer mit einem NULL-Zeichen (‘\0’).

Beispiel: **"Hallo"**

```
char   b[6]  = "Hallo";                    //Länge>= Anzahl Buchstaben + 1
     oder
char   b[]  = {‘H’, ‘a’, ‘l’, ‘l’, ‘o’, ’\0’};
```



Zeichenketten

```
int main()
{
    //Folgende Definitionen sind gleichwertig:
    char string1[6] = "Hallo";      //Das ist die bevorzugte Schreibweise
    char string2[] = "Hallo";
    char string3[6] = { 'H', 'a', 'l', 'l', 'o','\0' };

    printf("%s\n", string1);          //Ausgabe einer Zeichenkette mit %s
    ...
    return 0;
}
```

StringDef

Ein char-Array, das eine Zeichenkette speichert, muss immer um mindestens ein Element länger sein als die Anzahl der Zeichen.

Ein-/Ausgabe von Zeichenketten

```
char name[6];          //name kann maximal 5 Buchstaben aufnehmen

//Eingabe mit scanf_s():
 printf("Geben Sie den Namen ein>\n");
 scanf_s("%s", name, sizeof(name));
  /*1)Ein zusätzliches Parameter Buffersize definiert die maximale Anzahl von Buchstaben
     welche name aufnehmen kann(inklusive das letzte NULL-Zeichen).
     Falls die Anzahl eingegebenen Zeichen > Buffersize-1, Nichts wird eingelesen.
    2)Achtung: scanf liest nur bis zum ersten Whitespace(leerzeichen, Tabulator,
      Zeilenwechsel) */
//Ausgabe:
printf("Der Name ist %s", name);
```

StringEinAusgabe.cpp

Eine ganze Zeile einlesen und ausgeben

```
char line[100];        //Eine Zeile

//Eingabe mit gets_s():
 printf("Geben Sie eine Zeile ein>\n");
 gets_s ("%s", line, sizeof(line));
   //Ein zusätzliches Parameter Buffersize definiert die maximale Anzahl von Buchstaben
   //welche line aufnehmen kann(inklusive das letzte NULL-Zeichen).
   //Falls die Anzahl eingegebenen Zeichen > Buffersize-1, gibt es ein
   //Laufzeitfehler(Programmabsturz).

//Ausgabe:
puts(zeile);
```

StringEinAusgabe.cpp

Übung 1 : Eingabe vom Passwort

Erstellen Sie ein C-Programm, welches ein Passwort einliesst:

1. Das Passwort besteht aus Gross- oder Klein-Buchstaben, Ziffern (und Bindestrich), max. 8 Stellen.

2. Nach der Eingabe wird überprüft, ob alles eingelesen wurde. Falls es nicht der Fall ist, wird dem Benutzer erneut aufgefordert, das Passwort einzugeben.

Hinweis: Studieren Sie die formatierte Eingabe scanf() im Buch *Grundkurs C, Seite 334-335.*

```
scanf_s("%[a-zA-Z0-9]", pwd, 9)
```

Die obige Formatierung legt fest:

- Kleinbuchstaben von a bis z, sowie Großbuchstaben von A bis Z werden akzeptiert.
- Die Eingabe ist auf max. 8 Zeichen beschränkt. Falls man 9 Zeichen eingibt, wird nichts eingelesen. scanf_s() retourniert in diesem Fall 0.

C Bibliotheksfunktionen: string.h

Beschreibung	Funktionsname in C	Funktionsname in VS(C++)
Stringlänge	strlen()	**strlen**()
String-Kopierung	strcpy()	**strcpy_s**()
Verketten von Strings	strcat()	strcat_s()
Vergleich von Strings	strcmp()	strcmp()

Arbeitsauftrag zu Bibliothek string.h

- Studieren Sie Kapitel 2.8 vom Skript (M411_T_1045_V3.2.pdf, ohne Angaben zu C#)

- Erstellen Sie ein Testprogramm um die aufgelisteten vier String-Funktionen zu testen:

1) **strlen**()
2) **strcpy_s**()
3) strcat_s()
4) strcmp()

- Weitere Funktionen finden Sie unter:

https://openbook.rheinwerk-verlag.de/c_von_a_bis_z/011_c_arrays_013.htm