 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Zeichenketten (Strings)	Seite 1 von 11

Zeichenketten(Strings) sind wichtige Bestandteile fast eines jeden Programms. Beispiele sind Eingabe von Namen oder Adressen, Ausgabe von Meldetexten oder Bearbeitung ganzer Buchseiten. Bei der Programmiersprache C gibt es keinen speziellen Datentyp.

Zeichenketten sind uns bisher nur als konstante Texte begegnet, welche in "" angegeben waren; z.B. als Formatstring in der printf()-Funktion:

```
printf("\nHallo");
```

Ein einzelnes Zeichen wird im Speicher als ASCII-Code abgelegt. Z.B.: Der Buchstabe ‚A‘ hat den Binärcode 01000001 oder 65 (dezimal) oder 0x41 (hexadezimal). Eine Zeichenkette ist eine Folge von ASCII-Zeichen. Auch Ziffern sind im ASCII-Code festgelegt. Die Ziffern 0...9 haben den ASCII-Code 0x30...0x39.

Strings haben aber eine unterschiedliche Länge. Es muss eine Vereinbarung darüber getroffen werden, wie die unterschiedlichen Längen behandelt bzw. erkannt werden. In der Programmiersprache C ist festgelegt, dass das Ende eines Strings mit dem Zeichen 0x00 abschließt (Stringende-Kennzeichnung). In vielen Fällen muss sich der Programmierer nicht um das Stringende-Zeichen kümmern, da viele Stringfunktionen diese Zeichen automatisch einfügen oder berücksichtigen.

Merke:

Ein String ist eine Folge von ASCII-Zeichen, welche mit dem Zeichen 0x00 abgeschlossen ist. Jedes einzelne Zeichen belegt ein Byte, ist also vom Typ ‚char‘.

Für Strings muss, wie für alle anderen Objekte auch, Speicherplatz deklariert werden. Sie bilden ein **Datenfeld (Array)**, wobei jeder Wert des Datenfeldes vom **Typ ‚char‘** ist.

Deklarationsbeispiel:

```
char text[10]; //reserviert Platz für 10 Zeichen
```

Für die Festlegung der Größe des Datenfeldes sind zwei Dinge zu beachten:

- Der reservierte Speicherplatz muss so groß gewählt werden, wie der String maximal lang sein kann. Bei Überschreiten der Stringlänge gibt der Compiler keine Fehlermeldung aus. Schlimmstenfalls kommt es zur Laufzeit zum Programmabsturz.
- Auch für die abschließende 0x00 muss Speicherplatz reserviert werden. Das heißt im obigen Beispiel darf der String maximal 9 Zeichen lang sein. Das 10. Zeichen muss für 0x00 freigehalten werden.

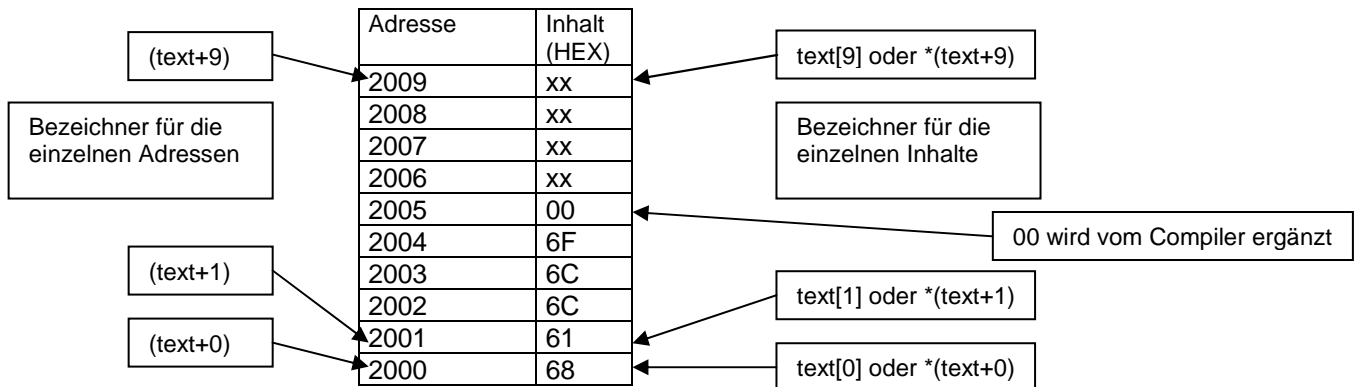
Der Bezeichner `text` ist ein Zeiger auf den Beginn des Strings. Der Zugriff auf die einzelnen Zeichen kann also wie bei Arrays auch mit der Zeigervariablen erfolgen.

Wie bei jeder Deklaration ist der Inhalt des Datenfeldes unbestimmt. Die Initialisierung kann auf verschiedene Arten erfolgen.

Deklaration mit Initialisierung:

```
char text[10]="Hallo";
```

Nach dieser Initialisierung könnte die Speicherbelegung folgendermaßen aussehen:



Wie bei allen Datenfeldern, kann auf die einzelnen Zeichen mit einem Index zugegriffen werden. Das erste Zeichen (ASCII von ‚H‘) trägt den Bezeichner `text[0]` (oder `*(text+0)`) usw..

Der Name ‚text‘ des Datenfeldes ist der Bezeichner für die Adresse, ab welcher der String im Speicher abgelegt ist. Mit Adressen kann man rechnen. Die folgenden Adressen sind `(text+1)`, `(text+2)`, usw. Die Adresse 2000 ist willkürlich gewählt und wird vom Compiler vergeben.

Die `strcpy_s()`-Funktion:

Zur Laufzeit des Programms lassen sich die Inhalte der Strings mit der Funktion

```
strcpy_s(text1, text2);
```


initialisieren. Die Übergabeparameter `text1` und `text2` sind dabei Adressen von Strings. Der Inhalt von `text2` wird nach `text1` kopiert. Es muss sichergestellt sein, dass der Zielstring mindestens die gleiche Stellenanzahl hat wie der Quellstring, sonst kommt es zu einem Fehler während der Laufzeit des Programms.

Beispiel:

```
char text[10];
strcpy_s(text, "Nepomuk") // "Nepomuk" wird ab Adresse text eingetragen
// 0x00 wird vom Compiler ergänzt.
```

oder:

```
char text1[10];
char text2[10]="Emil";
strcpy_s(text1, text2); // text1 bekommt den Inhalt "Emil"
```

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Zeichenketten (Strings)	Seite 3 von 11

In folgendem Beispiel wird ein Text1 durch einen anderen Text2 überschrieben:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    char text1[10]="Hallo";
    char text2[10]="Emil";
    printf("Text1: %s\n",text1);
    strcpy_s(text1,text2);
    printf("\nText1: %s\n",text1);
    _getch();
    return(0);
}
```

Anmerkung:

Die Zuweisung zu einem String mit dem '='-Operator ist nicht möglich!

Unzulässig ist beispielsweise:


```
text1="Nepomuk";//Das ist falsch!
```

Ergänzung:

Die Funktion `strcpy_s()` ist überladen. Das heißt, es gibt eine weitere Variante der Funktion mit drei Übergabeparametern:

```
strcpy_s(text1,buffer,text2);
```

Der zusätzliche zweite Übergabeparameter ist die Größe des Speicherplatzes (Buffer) für den Zielstring `text1`. Dieser Parameter muss angegeben werden, wenn die Adresszeiger rechnerisch bestimmt werden. Der Buffer entspricht der Länge von `text2` plus 1 (wegen dem zusätzlich nötigem Ende-Zeichen `0x00`). Die Ermittlung der Stringlänge mit der `strlen()`-Funktion wird weiter unten beschrieben.

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Zeichenketten (Strings)	Seite 4 von 11

Das folgende Beispiel hängt einen Teilstring (text2) an einen ersten String (text1) ab der 4. Position an. Dadurch wird text1 verlängert.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
int main()
{
    char text1[20]="Leo";
    char text2[10]="pold";
    printf("Text1: %s\n",text1);
    strcpy_s(text1+3,strlen(text2)+1,text2);
    printf("\nText1: %s\n",text1);
    _getch();
    return(0);
}
```

Ausgabe:

Text1: Leo

Text1: Leopold

Übung:

Ersetzen Sie in Zeile `strcpy_s(text1+3,strlen(text2)+1,text2);`
`...,text1+3,...` durch `text1+4!`
 und testen Sie das Programm! Begründen Sie bitte das neue Verhalten!

Die `gets_s()`-Funktion:


Die Tastatureingabe eines Strings zur Laufzeit des Programms erfolgt mit der Funktion

```
gets_s(text);
```

Der Parameter `text` ist dabei die Adresse des einzugebenden Strings.

Das folgende Beispiel erwartet die Eingabe eines Textes und gibt diesen Text anschließend wieder aus.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char text[100];
    printf("\nText: ");
    gets_s(text);
    printf("Eingegebener Text: %s\n",text);
    _getch();
    return(0);
}
```

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Zeichenketten (Strings)	Seite 5 von 11

Die strlen()-Funktion:


Die Funktion

```
int strlen(text);
```

berechnet die Länge eines Strings (das ist die Anzahl der Zeichen **ohne die abschließende 0**) und gibt den Wert als ganze Zahl zurück. Der Parameter `text` ist wieder die Adresse des Strings. Bei dieser Funktion muss die Header-Datei `string.h` eingebunden werden.

Beispiel:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>      //notwendig für die strlen()-Funktion
#include<stdlib.h>     //notwendig für die system()-Funktionen
int main()
{
    char text[100];
    int laenge;
    do
    {
        system("cls"); //Bildschirm löschen
        printf("\nText: ");
        gets_s(text);
        laenge=strlen(text);
        printf("\nLaenge des Textes %i Zeichen!\n",laenge);
        printf("'ESC' = Ende...");
    }
    while(!_getch()!=0x1b); //wiederhole solange nicht die 'ESC'-Taste betätigt
    return(0);
}
```

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Zeichenketten (Strings)	Seite 6 von 11

Das folgende etwas ausführlichere Beispiel demonstriert die bisher geschilderten Zusammenhänge. Es zeigt die Ablage eines Strings als Datenfeld im Speicher. Nach der Eingabe eines Strings wird anschließend die Speicherbelegung mit Adressen und deren Inhalt in HEX- und ASCII-Form dargestellt.

Analysieren Sie bitte das Programm und testen Sie es!

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
int main()
{
    char text[10]="?????????";
    int i;
    do
    {
        printf("\nText eingeben: ");
        gets_s(text);
        printf("\nEingegebener Text: %s",text);
        printf("\nTextlaenge: %i\n",strlen(text));
        for (i=0;i<10;i++)
        {
            printf("\nAdresse: %u Hex-Code: %2X Zeichen: %c",text+i,*(text+i),*(text+i));
        }
        printf("\n\nNochmal? j/n");
    }
    while(!_getch()=='j'); //wiederhole, wenn die 'j'-Taste betätigt wird
    return(0);
}
```

Die Ausgabe sieht beispielsweise folgendermaßen aus:

Text eingeben: Hallo

Eingegebener Text: Hallo

Textlaenge: 5

```
Adresse: 6880536 Hex-Code: 48 Zeichen: H
Adresse: 6880537 Hex-Code: 61 Zeichen: a
Adresse: 6880538 Hex-Code: 6C Zeichen: l
Adresse: 6880539 Hex-Code: 6C Zeichen: l
Adresse: 6880540 Hex-Code: 6F Zeichen: o
Adresse: 6880541 Hex-Code: 00 Zeichen:
Adresse: 6880542 Hex-Code: FFFFFFFE Zeichen: ■
Adresse: 6880543 Hex-Code: FFFFFFFE Zeichen: ■
Adresse: 6880544 Hex-Code: FFFFFFFE Zeichen: ■
Adresse: 6880545 Hex-Code: FFFFFFFE Zeichen: ■
```

Nochmal? j/n

strcat_s()-Funktion:

Die `strcat_s()`-Funktion hängt einen String an einen anderen an. Für diese Funktion muss die Header-Datei `string.h` eingebunden werden.

```
strcat_s(text1, text2);
```

Text2 wird an **Text1** angehängt. **Text1** wird dadurch verlängert.

Beispiel:

```
#include<stdio.h>
#include<conio.h>
#include<string.h> //wird für strcat_s()-Funktion benötigt
int main()
{
    char text1[30]="Haupt";
    char text2[30]="bahnhof";
    printf("Text1: %s",text1);
    strcat_s(text1,text2);
    printf("\nText1: %s\n",text1); //Ausgabe: Hauptbahnhof
    _getch();
    return(0);
}
```

Die strcmp()-Funktion:

Die `strcmp()`-Funktion führt einen lexikografischen (alphabetischen) Vergleich zweier Strings aus.

```
strcmp(text1, text2);
```

`text1` und `text2` sind die zu vergleichenden Strings.

Das Ergebnis der Funktion ist:

<0	String1 ist kleiner als String2
0	String1 ist identisch mit String2
>0	String1 ist größer als String2

Beim Vergleich zweier Strings wird Zeichen für Zeichen im ASCII-Code verglichen, bis sich eine Abweichung ergibt.

Beispiele:

```
char text1[30]="Fritz";  
char text2[30]="FRITZ";  
char text3[30]="Frank";  
char text4[30]="fritz";
```

```
strcmp(text1,text2); → Ergebnis ist >0  
strcmp(text1,text3); → Ergebnis ist >0  
strcmp(text2,text3); → Ergebnis ist <0  
strcmp(text2,text4); → Ergebnis ist <0
```

Testen Sie bitte die Beispiele mit folgendem Programm!


```
#include<stdio.h>  
#include<conio.h>  
#include<string.h>  
int main()  
{  
    char text1[30];  
    char text2[30];  
    int v;  
    do  
    {  
        printf("\nText1: ");gets_s(text1);  
        printf("\nText2: ");gets_s(text2);  
        v=strcmp(text1,text2);  
        if (v<0) printf("\n%s ist kleiner als %s!",text1,text2);  
        if (v>0) printf("\n%s ist groesser als %s!",text1,text2);  
        if (v==0) printf("\n%s ist gleich %s!",text1,text2);  
        printf("\n'ESC' = Ende... ");  
    }  
    while(_getch()!=0x1b);  
    return(0);  
}
```

Ausgabebeispiel:

Text1: Fritz

Text2: Frank

Fritz ist groesser als Frank!
'ESC' = Ende...

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Zeichenketten (Strings)	Seite 9 von 11

1. Übung:


Beschreiben Sie bitte die Wirkungsweise folgender Beispielprogramme bevor Sie diese testen!

Beispiel 1:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    char text1[30]="Donauschiffahrt";
    char text2[30]="dampf";
    printf("Text1: %s",text1);
    strcat_s(text2,text1+5);
    strcpy_s(text1+5,strlen(text2)+1,text2);
    printf("\nText1: %s\n",text1);
    _getch();
    return(0);
}
```

Beispiel 2:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    char text1[30]="Donau";
    char text2[30]="schiffahrt";
    int i=0;
    int j=strlen(text1);
    do
    {
        text1[j+i]=text2[i];
        i++;
    }
    while(text2[i]!=0);
    printf("\nText1: %s\n",text1);
    _getch();
    return(0);
}
```

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Zeichenketten (Strings)	Seite 10 von 11

Beispiel 3:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    char text[100];
    printf("Bitte Text eingeben: ");
    gets_s(text);
    int i=0;
    while(text[i]!=0) i++;
    printf("\nLaenge = %i Zeichen\n",i);
    _getch();
    return(0);
}
```

Beispiel 4:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    char text[30]="Donaudampfschiffahrt";
    printf("Text1: %s\n",text);
    strcpy_s(text+5,strlen(text)+1,text+10);
    printf("\nText1: %s\n",text);
    _getch();
    return(0);
}
```

2. Übung:

Schreiben Sie ein Programm, welches die Eingabe eines Textes gestattet und in einem Datenfeld ablegt.

Anschließend soll ein String erzeugt werden, in welchem der eingegebene Text in rückwärtiger Reihenfolge abgelegt und anschließend ausgegeben wird.


Die Eingabe soll solange erneut erfolgen, bis die ESC-Taste betätigt wird.

Beispiel:

Text: papier

Text rueckwaerts: reipap

'ESC' = Ende...

 - Lernmittel für moderne Technologien -	C/C++ - Programmierung	© Udo John www.lmt-verlag.de
	Zeichenketten (Strings)	Seite 11 von 11

3. Übung:

Entwickeln Sie ein Programm, welches die Zeichen eines eingegebenen Textes in Großbuchstaben wandelt und ausgibt!

4. Übung:

In den Rechner soll ein beliebiger Text und anschließend ein einzelnes Zeichen eingegeben werden. Entwickeln Sie ein Programm, welches feststellt, wie oft dieses Zeichen im Text vorkommt!

Beispiel:

Text: Dies ist ein Text

Zeichen: e

Das Zeichen e ist im Text 3 mal vorhanden!

'ESC' = Ende...

5. Übung:

Passwörter sollen i.a. bestimmte Anforderungen erfüllen (Sicherheitsrichtlinie). Erstellen Sie ein Programm, welches nach Eingabe eines Textes (Passwort) folgende Prüfung vornimmt:

- Das Passwort soll mindestens 8 Zeichen lang sein.
- Das Passwort soll jeweils mindestens einen Groß- und Kleinbuchstaben enthalten.
- Das Passwort soll mindestens einen Zahlenwert enthalten.
- Das Passwort soll mindestens eines der folgenden Sonderzeichen enthalten: ! _ & ? \$.

Das Programm liefert als Ergebnis eine Meldung 'Passwort korrekt' oder 'Passwort falsch'.