Zeichen (Character) und Zeichenkette (String)

Zur Speicherung eines Zeichens (Typ **char**) steht ein Byte zur Verfügung. Es können also 2^8 = 256 verschiedene Zeichen gespeichert werden. Jedes Zeichen ist durch eine Zahl codiert.

Aufgabe 1:
Mit folgendem Programm kann man sich den gesamten Zeichensatz und die dazugehörige Codierung anzeigen lassen. Testen Sie das Programm. Notieren Sie sich die Code-Nummern für die Buchstaben Ihres Familiennamens.

```
#include <iostream.h>
int  main()
{   char zeichen;
    for (int i = 0; i < 256; i++)
    {       zeichen = i;
            cout << i << ": " << zeichen << " ";
            if ((i % 10) == 0) cout << "\n";
    }
    return 0;
}
```

- **Zeichenketten** sind Felder aus Komponenten vom Char-Typ.
- Ein String wird mit ‚\0' beendet (die kürzeste Zeichenkette besteht also mindestens aus einem Zeichen).
- Zeichenketten werden in doppelte Anführungszeichen eingeschlossen.
- Über das Gleichheitszeichen kann man einer Variablen keine Zeichenkette zuweisen. Für diesen Fall stehen uns spezielle String-Funktionen in der Bibliotheksdatei ***string.h*** zur Verfügung.
- ***strlen(s)*** liefert Länge des Strings *s*
- ***strcpy(s1, s2)*** kopiert *s*2 auf *s*1 (einschließlich \0)
- ***strcat(s1, s2)*** fügt *s*2 an *s*1 an (\0 von *s*1 wird überschrieben), Funktionswert ist *s*1
- ***strcmp(s1, s2)*** vergleicht *s*1 mit *s*2, Wert ist <, =, > 0, falls *s*1 <, =, > *s*2.

Beispiel:

```
#include <iostream.h>
int main()
{       char wort[10];              //Vereinbarung einer Zeichenkette der Länge 10
        wort[0] = 'H'; wort[1] = 'a'; wort[2] = 'l'; wort[3] = 'l'; wort[4] = 'o';  wort[5] = '\0';
        cout << "\nDas Wort ist: " << wort << endl;
        return 0;
}
```

Aufgabe 2:
Schreiben Sie ein Programm, das zunächst eine Zeichenkette ***name*** definiert, die so lang wie Ihr Familienname ist, anschließend den Komponenten die Buchstaben Ihres Familiennamens mit Hilfe ihrer Codierung durch die entsprechenden Dezimalzahlen zuweist und den Namen auf dem Bildschirm ausgibt.

Aufgabe 3:
Erstellen Sie ein Programm, das einer Stringvariablen den Wert *Weihnachts* zuweist (mit *strcpy*), in das ein weiterer String über Tastatur (mittels *cin*) eingegeben wird (z. B. *mann* oder *gans*), das beide Strings verbindet und den Ergebnis-String und dessen Länge ausgibt.

Anmerkung: Gibt man mit *cin* eine Zeichenkette ein, die Leerzeichen enthält, dann wird die Zeichenkette nur bis zu diesem Abbruchzeichen erfasst. Soll eine Zeichenkette Leerzeichen enthalten, so hat man die Möglichkeit die Eingabe mit Hilfe der Methode ***get*** zu realisieren:
cin.get(wort, 20); //Eingabe der Zeichenkette *wort* mit max. Länge von 20 Zeichen

Aufgabe 4:
Erstellen Sie ein Programm, das ein zweidimensionales Feld aus Leerzeichen und dem Zeichen * erzeugt, so dass bei der Ausgabe der Zeichen auf den Bildschirm ein „Weihnachtsbaum“ etwa wie folgt „gezeichnet“ wird.

```
   *
  * *
 *   *
*******
   *
   *
```

Hinweis:
Feldvereinbarung z. B. durch ***char baum[6][8];***
Wertzuweisung für die 3. Zeile des Feldes ***strcpy(baum[2], “ * * \n“);***

Der strukturierte Datentyp *struct*

Ein ***struct*** (ein Datensatz, eine Struktur, ein Record) ist ein Datentyp mit einer **festen** Anzahl von Komponenten, von denen jede einen **beliebigen** Typ haben kann. Der **Zugriff** auf die Komponenten des Datensatzes erfolgt **direkt** durch Angabe des Struct- und des Komponentennamens getrennt durch einen Punkt.
Beispiel für die Deklaration eines Strukturtyps:

```
struct tDatum              //das ‚t’ soll an ‚Typ’ erinnern
{
        short tag;
        char monat[10];
        int jahr;
};                         //Semikolon muss sein!
```

Beispiele für Variablenvereinbarungen und Wertzuweisungen für diesen Strukturtyp:

```
tDatum gruendung = {17, “Oktober“, 1456};      //EMAU
tDatum mein;
```

Aufgabe 5:
Erstellen Sie ein Programm, das obige Vereinbarungen enthält, die Eingabe Ihres Geburtsdatums in die Variable *mein* ermöglicht und die Tage und Monate der beiden Daten wieder ausgibt. Hinweis: Ausgabe der Komponente *monat* der Struktur *gruendung* durch:
cout << ***gruendung.monat***;

Aufgabe 6:
Erstellen Sie ein Programm, das den Strukturtyp *tMitglied* deklariert, der die Namen und Geburtsdaten von Familienmitgliedern enthält und diese in einem Feld *familie* der Länge *n* (zunächst *n* = 3) speichert.