Funktionen mit Feldern als Parameter

Bislang wurde unterschieden:

- Eingabeparameter (Call-by-Value)
- Rückgabewerte (per return)
- Ein- und Ausgabeparameter (Call-by-Reference)

Mit Call-by-Reference werden Parameter als Zeiger auf Variablen übergeben.

Felder:

- Felder werden als Zeiger an eine Funktion übergeben

Funktionen mit Feldern als Parameter

Felder als Funktionsparameter:

- Übergebene Felder werden wie Zeiger behandelt.
- Ein Feld-Variablenname ohne Indexklammer ist eine Adresse
- Die Parameterübergabe erfolgt immer per Call-by-Reference

Aus diesem Grund kann bei Feldern nicht nach Ein- bzw. Ausgabeparametern unterschieden werden, da ohnehin die Adressen der Felder übergeben werden.

In der Parameterliste wird ein Feld durch den Typ der Elemente, den Namen und ein Paar eckige Klammern gekennzeichnet, oder alternativ als Zeiger auf den Typ der Feldelemente.

Beispiel mit eindimensionalen Feldern als Parameter

Funktion zur Berechnung des arithmetischen Mittels eines Feldes von float-Zahlen mit n Elementen:

Funktionsdefinition:

```
float amittel(int n, float feld[ ])  // auch float *feld möglich
{ int i;
  float s=0.0f;
  for (i=0;i<n;i++)
    s=s+feld[i]; // auch *(feld+i) möglich
  return s/(float)n;
}
```

Beachte: bei eindimensionalen Feldern kann ein leeres Klammerpaar [ ] angegeben werden.

Beispiel mit eindimensionalen Feldern als Parameter

Funktionsaufruf:

```
void main()
{  int anzahl;
   float mittel,messwerte[100];
   // Anzahl und Messwerte werden
   // durch Einlesen belegt
   mittel=amittel(anzahl, messwerte) ;

   printf("\narithm. Mittel=%f",mittel);
}
```

Beispiel mit mehrdimensionalen Feldern als Parameter

Funktion zum Transponieren einer quadratischen Matrix mit n^2 Elementen des Typs integer.

Funktionsdefinition:

```
void trans(int n, int ma[ ][10])
{ int i,j,h;
  for (i=0;i<n;i++)
    for (j=0;j<i;j++)
    {  h=ma[i][j];
       ma[i][j]=ma[j][i];
      ma[j][i]=h;
     }
}
```

Beachte: bei mehrdimensionalen Feldern müssen nach einem möglicherweise leeren ersten Klammerpaar [ ] (Anzahl der Zeilen) unbedingt die nachfolgenden Klammerpaare mit Konstanten, die die genaue Anzahl für diese Dimension darstellen (z.B. genaue Anzahl der Spalten usw.) angegeben werden.
Die Anzahl bezieht sich auf die Deklarationsgrenzen des übergebenen Argumentes, nicht auf die möglicherweise geringere Anzahl benutzter Elemente!

Beispiel mit mehrdimensionalen Feldern als Parameter

Funktionsaufruf:

```
void main()
{ int j,i,m;
  int x[10][10];
  printf("\nm=");
  scanf("%d",&m); // m<=10
  //Belegung der Matrix x
  for (i=0;i<m;i++)
    for (j=0;j<m;j++)
     {  printf("\nx[%d][%d]=",i,j);
        scanf("%d",&x[i][j]);
     };
```

```
    //Transponieren
    trans(m,x);

    for (i=0;i<m;i++)
      for (j=0;j<m;j++)
         printf("\nx[%d,%d]=%d",
                  i,j,x[i][j]);
}
```