

Programmieren mit Zeigern („Ortsangaben“)

Beispiel 2: „Übergabe“ eines Vektors an eine Funktion

```
// Die Funktion print_vec() gibt alle Elemente eines Vektors auf dem
// Bildschirm aus. Bitte beachten Sie, dass es an verschiedenen Stellen
// im Programm zwei mögliche Schreibweisen mit derselben Bedeutung gibt.
// Sie müssen sich jeweils für eine der beiden Varianten entscheiden!
#include <stdio.h>
```

Erster Parameter: Ortsangabe, hier soll der Ort des nullten Vektor-Elements übergeben werden.

```
// Deklaration
void print_vec(int* vec, int anz); // Entweder Variante 1
void print_vec(int vec[], int anz); // oder die Variante 2
```

```
int main(void)
{
    int daten[5] = { 55, 44, 33, 22, 11 };

    // Funktionsaufruf: Es wird der Ort des nullten Vektor-Elements
    // im Hauptspeicher des Rechners übergeben. Zusätzlich wird die
    // Zahl der auszugebenden Elemente übergeben.
    print_vec(&daten[0], 5); // Entweder Variante 1
    print_vec( daten, 5); // oder die Variante 2

    return 0;
}
```

&daten[0] bedeutet: Ort des nullten Vektor-Elements im Hauptspeicher des Rechners. Die Kurzschreibweise (Variante 2) hat dieselbe Bedeutung!

```
// Definition
void print_vec(int* vec, int anz) // Entweder Variante 1
void print_vec(int vec[], int anz) // oder die Variante 2
{
    int i;
    for(i = 0; i < anz; i++)
    {
        // Ausgabe des i-ten Vektor-Elements.
        printf("%d\n", *(vec+i) ); // Entweder Variante 1
        printf("%d\n", vec[i] ); // oder die Variante 2
    }
}
```

vec ist der Ort des nullten Vektor-Elements, (vec+i) ist der Ort des i-ten Vektor-Elements, *(vec+i) ist das i-te Vektor-Element, welches hier ausgegeben wird.