

## 4.2 Vektortypen (*array types*)

erlauben die Vereinbarung von Feldern (*arrays*). Ein Feld besteht aus einer festen Anzahl gleichartiger Elemente. Der Typ eines Feldes (Vektortyp) beschreibt Anzahl und Datentyp der Feldelemente.

### 4.2.1 Eindimensionale Felder: Vektoren

Syntax der Vereinbarung:

*type identifier* [ *const\_expr* *opt* ]

*type*          Datentyp der Vektorelemente

*identifier*    Name des Vektors

*const\_expr* • legt die Anzahl der Vektorelemente fest (Vektorlänge)  
• darf z.B. weggelassen werden, wenn die Größe des Vektors durch die Anzahl der Initialisierungswerte festgelegt wird.

Die Vektorlemente werden in einem zusammenhängenden Speicherbereich abgelegt. Der **Index-Operator** [ ] ermöglicht den Zugriff auf einzelne Vektorelemente:

*identifier*[0] . . . *identifier*[*const\_expr*-1]

Beispiel:

```
#define VEKLEN 6

int main( void ) {
    short int a[VEKLEN], sum[VEKLEN];
    short int b[] = {1, 2, 3, 4, -1};
    int i;

    for (i = 0; i < VEKLEN; i++)
        a[i] = i;
    for (i = 0; (i < VEKLEN) && (b[i] >= 0); i++)
        sum[i] = a[i] + b[i];
    while (i < VEKLEN) {
        sum[i] = i;
        i++;
    }

    /* a:      0  1  2  3  4  5  */
    /* b:      1  2  3  4 -1  */
    /* sum:    1  3  5  7  4  5  */
}
```

**Bereichsüberschreitung:**

- liegt vor, wenn beim Zugriff auf *identifier*[*i*] der Index *i* außerhalb der zulässigen Grenzen liegt, also *i* < 0 oder *i* ≥ *const\_expr* gilt.
- häufige Ursache für Laufzeitfehler.

**Zeichenkettenkonstante (String):**

- **konstanter** Vektor mit Elementen des Typs **char**
- darf indiziert, jedoch nicht verändert werden:

```
"abc"[1] → 'b'
"abc"[1] = 'B'; /* unzulässig */
```

## 4.2.2 Zweidimensionale Felder: Matrizen

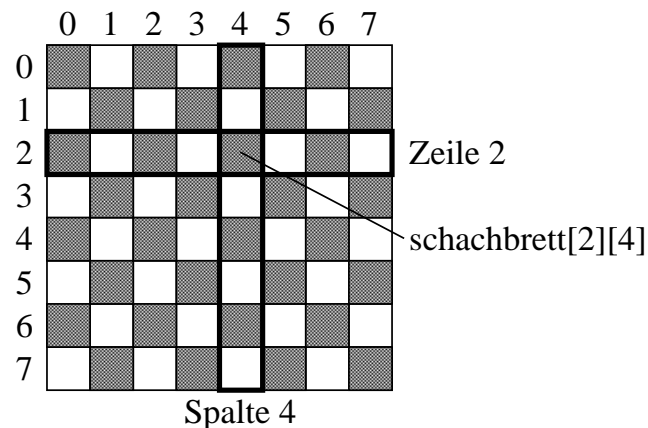
Syntax der Vereinbarung:

```
type identifier[ const_expr1opt ][ const_expr2]
```

*type*            Datentyp der Matrixelemente  
*identifier*     Name der Matrix  
*const\_expr<sub>1</sub>*    Anzahl der **Zeilen**  
*const\_expr<sub>2</sub>*    Anzahl der **Spalten**

Die Elemente einer Matrix werden zeilenweise in einem zusammenhängenden Speicherbereich angeordnet. Die Vorstellung eines zweidimensionalen Feldes entspricht dagegen einem rechteckigen Bereich mit Zeilen und Spalten.

Ein Beispiel für eine Matrix ist ein Schachbrett:



## Zugriff auf ein Matrixelement

erfolgt durch zweifache Indizierung:

```
identifier[zeile][spalte]
```

wobei für die Indizes *zeile* und *spalte* gelten muß:

$$0 \leq \text{zeile} < \text{const\_expr}_1$$
$$0 \leq \text{spalte} < \text{const\_expr}_2$$

## Beispiel:

```
char schachbrett[8][8];  
int z, s;  
  
for (z = 0; z < 8; z++)  
    for (s = 0; s < 8; s++)  
        if ( (z+s)%2 == 0 )  
            schachbrett[z][s] = 'X';  
        else  
            schachbrett[z][s] = ' ';
```