






## ▪ \*ptr++

-  `ptr` sei ein Zeiger auf den Anfang eines Feldes von Ganzzahlen, z.B. `{ 0, 0, 0 }`
-  `(*ptr)++`: Dereferenziere `ptr` und inkrementiere die erste Zahl => `{ 1, 0, 0 }`
-  `*(ptr++)`: Setze `ptr` um eine Zahl weiter (zeigt anschließend auf die zweite Zahl des Feldes), dereferenziere den bisherigen Zeiger und vergesse das Ergebnis => `{ 0, 0 }`

## ▪ \*argv[1]

- **Beispiel:** `argv = { "abc", "def", "ghi" }`
-  `(*argv)[1]`: Liefert das zweite Zeichen der ersten Zeichenfolge ('b')
-  `*(argv[1])`: Liefert das erste Zeichen der zweiten Zeichenfolge ('d')

## ▪ a & b statt a && b: Funktioniert aufgrund des schwachen Typsystems, aber...

- ... Ergebnis für `&` hängt von Bitpositionen ab
- ... keine verkürzte Auswertung mit garantierter Reihenfolge beim `&`

```
int a = 1, b = 2;
if( a & b ) { ... }    /* 0 bzw. false */
if( a && b ) { ... }   /* 1 bzw. true  */
(fprintf(stdout, "Left"), 0) & (fprintf(stdout, "Right"), 0); /* > LeftRight */
(fprintf(stdout, "Left"), 0) && (fprintf(stdout, "Right"), 0); /* > Left */
```