

Aufgabe 4: Zeiger

Zeiger (pointer) sind eine vielseitige Möglichkeit auf Speicherinhalt zuzugreifen. Ein Zeiger ist eine Variable in der man Speicheradressen ablegen kann. Ein Zeiger muss in der Regel einen Typen haben, der zu der Variable passt, auf die er zeigt.

Möchte man einen Zeiger von Typ `int` anlegen muss man den `*`-Operator verwenden:

```
int * zeiger;
```

Mit dem `&`-Operator kann man auf die Adresse einer Variable zugreifen. Dies ist notwendig, wenn man die Adresse einer Variable einem Zeiger mitgeben möchte:

```
int wert;  
  
wert = 5;  
zeiger = &wert;
```

Mit dem `*`-Operator kann man nun auf den Inhalt des Zeigers zugreifen:

```
*zeiger = *zeiger + 1;  
printf("Wert auf den der Zeiger zeigt: %d", *zeiger);
```

Es wird nun folgendes ausgegeben:

```
Wert auf den der Zeiger zeigt: 6
```

Das macht noch nicht viel Sinn. Man hätte dafür auch die normalen Variablen verwenden können. Es wird schon interessanter, wenn ein Zeiger auf einen Arrayeintrag zeigt:

```
int mein_array[] = {1,23,17,4,-5,100};  
int * zeiger;  
int i;  
  
zeiger = &mein_array[0];
```

Man kann mit Zeigern rechnen:

```
for (i = 0; i < 6; i++)  
{  
    printf("Zeigerwert %d : %d \n", i, *zeiger);  
  
    zeiger++; /* hier wird der Zeiger inkrementiert */  
}
```

Aufgabe:

1. Sortiere das oben genannte Feld (`mein_array`) aufsteigend nach Größe.
2. Benutze dafür Zeiger.

3. Es darf kein zweites Feld verwendet werden. Also so wenig Speicher wie möglich verbrauchen.

[Hilfe zu Zeigern](#)