

Modellierung und Programmierung – Aufgabenserie 2b

Aufgabe 3:

```
1 char txt[] = "abcdefghij";
2 char *zeiger = &txt[1];
3
4 printf("%p\n", txt);
→ 0x7fff56370bcd, da %p den Zeiger auf txt ausgibt. Also wird die
   Speicheradresse von text ausgegeben.
5 printf("%p\n", &zeiger[4]);
→ 0x7fff56370bd2, da zeiger auf &txt[1] zeigt, also eine Adresse weiter, als
   txt. Also zeigt &zeiger[4] auf &txt[5], also noch einmal vier Adressen
   weiter.
6 printf("%c\n", *zeiger);
→ b, da der zeiger mit * dereferenziert wird, also der Wert an txt[1]
   ausgegeben wird.
7 printf("%p\n", zeiger + 3);
→ 0x7fff56370bd1, da zeiger auf &txt[1] zeigt. Werden also 3 dazu addiert,
   erhöht sich die Adresse bezüglich der Adresse von txt um 4.
8 printf("%c\n", *(zeiger + 1 - 1));
→ b, da zeiger + 1 - 1 = zeiger, also dasselbe wie in Zeile 6
9 printf("%p\n", ++zeiger);
→ 0x7fff56370bcf, da ähnlich wie in Zeile 7 nur der Wert der Adresse des
   Zeigers um 1 erhöht wird. Diesmal allerdings langfristig.
10 printf("%c\n", +++zeiger);
→ d, da der Wert von zeiger zuerst erneut erhöht wurde, und deswegen nun nicht
   mehr auf b, sondern zwei Stellen weiter im txt-Array auf d zeigt.
11 printf("%p\n", zeiger);
→ 0x7fff56370bd0, da der Zeiger, wie gesagt, um zwei Adressen verschoben wurde.
12 printf("%c\n", --(*zeiger));
→ c, da nun der Wert des Objekts, auf das zeiger zeigt (txt[3]), um 1
   verringert wurde und dessen ASCII-Code nun dem c zugeordnet ist.
13 printf("%c\n", *zeiger++);
→ c, da die Adresse, auf die zeiger zeigt, erst nach dieser Ausgabe erhöht
   wurde.
14 printf("%c\n", *(&zeiger[1] - 1));
→ e, da zeiger nun e zeigt, also &zeiger[1] auf f. Wird die Adresse von f aber
   um 1 verringert, ist man wieder bei e.
15 printf("%c\n", *(zeiger - 3));
→ b, da der zeiger auf e zeigt, und drei Adressen davor b steht.
16 printf("%c\n", *zeiger);
→ e, da der zeiger dreimal um 1 erhöht wurde und deswegen nun auf txt[4] zeigt
17 printf("%c\n", zeiger[-1]);
→ c, da der Wert an der Adresse vor e, also dem Wert, auf den der Zeiger zeigt
   in Zeile 12 verringert wurde.
18 printf("%c\n", (zeiger + 1)[-1]);
→ e, da erst die Adresse des Zeigers um 1 erhöht wird, dann aber im Feld wieder
   einen Schritt zurückgegangen wird.
19 printf("%ld\n", zeiger - txt);
→ 4, da der zeiger auf die Stelle txt[4] zeigt.
```