

Programmieren C: Einfache rekursive Funktionen: Rekursive Ausgabe einer Dezimalzahl

Klaus Kusche

Das Umwandeln eines **int** in eine Folge von Ziffern geschieht am einfachsten von hinten nach vorne: Man dividiert die Zahl durch 10; der Divisionsrest ist die nächste Stelle von hinten nach vorne, und mit dem Divisionsergebnis rechnet man weiter. Das wiederholt man, bis die Zahl 0 ist.

Leider kann man nicht von hinten nach vorne ausgeben, und auch das direkte Speichern von hinten nach vorne in einem String ist nur möglich, wenn man vorab festlegt, wie viele Zeichen die Zahl im String schließlich belegen soll. Wenn man hingegen nur so viele Zeichen belegen will, wie die Zahl wirklich lang ist, muss man die Ziffern entweder aufwändiger von vorne nach hinten ausrechnen, oder man muss das Ergebnis zwischenspeichern und dann umdrehen oder verschieben.

Hier hilft die Rekursion, und zwar in Form folgender Idee für eine Funktion, die einen **int** genau so breit wie nötig als Dezimalzahl ausgibt¹:

- Wenn die Zahl größer als 9 ist, werden zuerst die vorderen Stellen (d.h. die Zahl mit der letzten Ziffer wegdividiert) mit einem rekursiven Aufruf ausgegeben.
- Zuletzt wird die letzte Ziffer der Zahl als einzelnes Zeichen ausgegeben.

Schreib eine Funktion, die genau das macht (die Funktion hat keinen Returnwert), und ein Hauptprogramm dazu, das alle Worte der Befehlszeile mittels **atoi** in **int**'s verwandelt und mit dieser Funktion einzeln zeilenweise ausgibt.

Als einzige Ausgabe-Funktion steht dir **putchar(c)** aus **stdio.h** zur Verfügung, wobei *c* ein einzelnes Zeichen (bzw. ein ASCII-Wert als Zahl) ist (erinnere dich, wie man aus einem Ziffernwert zwischen 0 und 9 den ASCII-Wert des Ziffern-Zeichens berechnet!).

Zusatzaufgabe:

- Erweitere deine Funktion so, dass sie auch negative Zahlen richtig ausgibt.

¹ Diese Idee ist zwar die eleganteste, aber nicht unbedingt die effizienteste!