

# Programmieren C: Funktionen und Arrays: Pascal'sches Dreieck

## Klaus Kusche

Wir schreiben nochmals ein Programm, das das Pascal'sche Dreieck ausgibt (siehe vorangegangene Übungen), aber diesmal zeit- und platzeffizient:

Ohne komplizierte oder rekursive Einzel-Berechnung der Binominalkoeffizienten, und nur mit einem eindimensionalen (statt zweidimensionalen) Array.

Außerdem soll das Programm in Funktionen aufgeteilt werden:

- Zuerst brauchen wir eine ganz einfache Hilfsfunktion für die Ausgabe einer Zeile: Sie wird mit einem **int-Array** und der Anzahl der daraus auszugebenden Zahlen aufgerufen und hat keinen Returnwert.

Die Zahlen sollen in einer Zeile nebeneinander ausgegeben werden (mit jeweils einem Zwischenraum dazwischen und einem Zeilenvorschub am Ende, es geht uns diesmal nur um die Funktion, nicht um die schöne Ausgabe).

- Die eigentliche Berechnungsfunktion hat keinen Returnwert und einen Parameter: Die Anzahl der auszugebenden Zeilen.

Sie legt zuerst einmal ein Array für die jeweils aktuelle Zeile des Dreiecks an. Die unterste Zeile des Dreiecks enthält so viele Zahlen wie es Zeilen sind, also muss das Array auch diese Größe haben.

Dann befüllt sie das Array mit den Werten der ersten Zeile: Vorderstes Element **1**, alle anderen Elemente **0**.

Danach ruft sie die Ausgabe-Funktion mit dem Array auf, um die erste Zeile (nur 1 Wert!) auszugeben.

Es folgt eine Schleife, die der Reihe nach die zweite bis letzte Zeile berechnet und gleich ausgibt (wie viele Zahlen werden in der i-ten Zeile ausgegeben?).

Für die Berechnung der neuen Zeile aus der alten Zeile gilt:

- Für **a[0]** braucht man nichts tun: Es enthält immer den Wert **1** und muss daher nicht neu berechnet werden.
- Für die Berechnung des neuen Wertes der restlichen **a[i]** muss man die bisherigen Werte von **a[i-1]** und **a[i]** zusammenzählen.

Würde man von links nach rechts arbeiten, bräuchte man zwei Arrays (eines für die alte und eines für die neue Zeile), weil man das alte **a[i]** noch zur Berechnung des neuen **a[i+1]** braucht und daher nicht mit dem neuen **a[i]** überschreiben darf.

Wenn man die neuen Werte aber von hinten nach vorne berechnet (von rechts nach links), kann man sie direkt statt den alten Werten im selben Array speichern und braucht daher nur ein einziges Array.

- Das Hauptprogramm wird mit einer Zahl auf der Befehlszeile aufgerufen: Der Anzahl der Zeilen, die auszugeben sind. Es prüft nur die Eingabe (sollte größer 0 sein!) und ruft die Pascal-Funktion auf.