

# Programmieren C: **switch**: Ganz einfacher Taschenrechner

## **Klaus Kusche**

Schreib ein Programm, das ganz einfache Rechnungen ausrechnet:

- Die Rechnungen bestehen aus Kommazahlen und + - \* / ^ (^ für "hoch")<sup>1</sup>.
- Es wird einfach Rechenzeichen für Rechenzeichen von links nach rechts gerechnet, ganz ohne Vorrangregeln und ohne Klammern<sup>2</sup>.

### Aufruf:

- Die Rechnung wird auf der Befehlszeile angegeben, das Endergebnis soll ausgegeben werden.
- Jede Zahl und jedes Rechenzeichen ist ein einzelnes Eingabe-Wort (d.h. du musst Zwischenräume vor und nach jedem Rechenzeichen eingeben!).

### Hinweise:

- Das Programm soll mit **double**-Zahlen rechnen.
- Verwende für die Unterscheidung der 5 Rechenzeichen **switch** und **case** !  
Bei ungültigen Rechenzeichen sollst du eine Fehlermeldung ausgeben und abbrechen.
- Für ^ : Die Funktion heißt **pow** und kommt aus **math.h**<sup>3</sup>.
- Prüfe zu Beginn des Programms auf die richtige Anzahl von Eingabe-Worten:  
1, 3, 5, ... Worte (ohne Programmnamen) sind ok,  
0, 2, 4, ... können keine gültige Rechnung sein. (Warum?)

### Tipp:

- Verwandle als erstes das vorderste Eingabe-Wort in eine Zahl und speichere diese als vorläufiges Zwischenergebnis.
- Mach dann eine Schleife in Zweiserschritten über die restlichen Eingabe-Worte:
  - Lies zuerst einmal das hintere der beiden Worte als neue Zahl.
  - Im ersten Zeichen des vorderen Wortes muss das Rechenzeichen stehen:  
Mit **argv[i][j]** bekommst du das **j**-te Zeichen (einen **char**) im **i**-ten Wort (auch **j** beginnt bei 0).
  - Mach eine Fallunterscheidung (switch !) je nach Rechenzeichen und rechne die neue Zahl dementsprechend zum bisherigen Zwischenergebnis dazu (prüfe beim Dividieren auf 0!).

- 1 **Achtung:** Wenn du unter Linux arbeitest, musst du auf der Befehlszeile \\* statt \* eintippen! Auch unter DOS/Windows müssen \* und ^ speziell eingegeben werden. Am einfachsten ist es, in diesem Programm **x** statt \* und **p** statt ^ zu verwenden!
- 2 Die Berücksichtigung von Klammern ist zu schwierig, die Beachtung der Vorrangregeln ist noch schwieriger (kommt bestenfalls in einem Wahlpflichtfach des Studiums). Für beides braucht man sinnvollerweise Funktionen und Rekursion (==> später!).
- 3 Für Linux und Mac: Option **-lm** beim Kompilieren bzw. Linken nicht vergessen!