Programmieren C: Schleifen, Rechnungen, “bis ungefähr gleich”: Näherungs-Berechnung der Quadratwurzel

Klaus Kusche

Jetzt verwenden wir Kommazahlen statt ganze Zahlen:

- Der Typ dafür heisst **`double`** statt **`int`**.
- Der **`printf`**-Platzhalter heisst **`%f`** statt **`%d`**
  (bzw. z.B. **`%.15f`**, wenn du 15 Kommastellen sehen willst).
  Für besonders große oder kleine Zahlen ist **`%.15g`** besser.
- Und das Umwandeln von Text auf Kommazahl funktioniert mit **`atof`** statt **`atoi`**.

Die Aufgabe ist folgende:

“Ermittle die Wurzel einer gegebenen Zahl nach folgendem schrittweisen Näherungsverfahren:

- *Beginne mit der Zahl selbst als erstem Näherungswert.*
- *Berechne in jedem Durchlauf den neuen Näherungswert als Mittelwert des alten Näherungswertes und des Divisionsergebnisses von “ursprünglicher Zahl durch alten Näherungswert”.*
- *Wiederhole das, bis das Quadrat des Näherungswertes erst nach der zehnten Stelle von der ursprünglichen Zahl abweicht (anders ausgedrückt: Wiederhole das, bis der Absolutwert von (Quadrat des Näherungswertes - ursprüngliche Zahl) kleiner ist als (ursprüngliche Zahl * 10^{-10})).”*

Dieses Verfahren zur Berechnung von Wurzeln ist das seit Jahrtausenden bekannte “Verfahren von Heron”, siehe z.B. https://de.wikipedia.org/wiki/Heron-Verfahren .

Beim Prüfen auf das Milliardstel Abweichung wird die Bedingung einfacher, wenn du die Funktion **`fabs`** verwendest (wie **`abs`**, aber für **`double`**: Absolutbetrag, d.h. das Ergebnis ist der hineingesteckte Wert ohne Vorzeichen).
Für **`fabs`** gilt: Ein **`double`** geht rein, ein **`double`** kommt raus, kommt aus **`math.h`** .

Tipps / Zusatzaufgaben:

- Zum Vergleich kann dein **`printf`** auch das Ergebnis der *eingebauten Wurzelfunktion* ausgeben: Die Funktion heißt **`sqrt`** (ein **`double`** geht rein, und das Ergebnis ist auch ein **`double`**), und sie kommt aus **`math.h`** .

  Hinweis für Linux- und Mac-User:
  Alle *mathematischen Funktionen* (z.B. **`sqrt`**) sind unter Unix bzw. Linux aus historischen Gründen in einer *separaten Library*. Zum Einbinden dieser Library braucht man beim Kompilieren bzw. Linken auf der Befehlszeile zusätzlich **`-lm`** .
- Für Neugierige: Kannst du die *Anzahl* der Näherungsschritte ausgeben? Den *Näherungswert nach jedem Schritt*?
- Und die übliche Frage: *Wie deppensicher* ist dein Programm? (Was kann bei der *Eingabezahl* für die Wurzel *schiefgehen*?)