

*Schleifen, Rechnen mit ganzen Zahlen*

**Wichtig:** Erstellen Sie vor dem Programmieren jeweils ein Struktogramm!

- 1) Erstellen Sie ein Programm zur Berechnung der n-ten Potenz einer ganzen Zahl durch fortgesetzte Multiplikation (z. B.  $6^3 = 6 \cdot 6 \cdot 6$ )!  
Verwenden Sie nur die 4 Grundrechenarten, keine mathematischen Bibliotheksfunktionen!  
Bei der Eingabe ist sicherzustellen, dass der Exponent n nicht negativ ist!  
Schreiben Sie das Programm in drei Formen, und zwar unter Verwendung
  - (a) einer fußgesteuerten Schleife,
  - (b) einer kopfgesteuerten Schleife mit `while` und
  - (c) einer kopfgesteuerten Schleife mit `for` !
  
- 2) Erstellen Sie basierend auf (1c) ein Programm, das das Gleiche leistet mit folgenden Erweiterungen:  
Als Basis sind reelle Zahlen möglich und für den Exponenten sind beliebige ganze Zahlen erlaubt.
  
- 3) Auf einen Sparvertrag wird n Jahre lang zu Beginn des Jahres die Sparrate s eingezahlt. Der jährlicher Zinssatz des Sparvertrages bleibt konstant p %.  
Erstellen Sie eine Tabelle, aus der für jedes Jahr der Laufzeit die in diesem Jahr erhaltenen Zinsen und das Gesamtguthaben am Jahresende zu ersehen sind!  
Die Laufzeit n (in Jahren), die jährliche Sparrate s (in €) und der Zinssatz p (in %) sind vom Benutzer zu erfragen!  
Hinweis:  
Das Guthaben am Ende eines Jahres setzt sich zusammen aus
  - dem Guthaben am Ende des Vorjahres,
  - der zu Beginn des Jahres eingezahlten Sparrate,
  - den Zinsen auf das während des Jahres vorhandene konstante Guthaben.
  
- 4) Berechnen Sie den natürlichen Logarithmus  $\ln x$  für eine vom Benutzer einzugebende Zahl x mit  $0 < x \leq 2$  nach der folgenden Reihenentwicklung:  
$$\ln x = (x - 1) - \frac{(x - 1)^2}{2} + \frac{(x - 1)^3}{3} - \frac{(x - 1)^4}{4} + \dots$$
  
Berechnen Sie die Reihe ab, wenn der Betrag des letzten Summanden kleiner als eine vorgegebene Zahl  $\varepsilon$  ist!  
Verwenden Sie bei Ihrer Berechnung keine mathematische Funktion!  
Die Funktion `log`, die mit `#include <math.h>` zur Verfügung steht, können Sie benutzen, um Ihr Ergebnis zu kontrollieren.

- 5) Erstellen Sie ein Programm, das einen einfachen Taschenrechner simuliert, in den abwechselnd eine Zahl und ein Operator (+, −, \*, /) für eine der vier Grundrechenarten eingegeben werden können.

Die Rechnung soll so durchgeführt werden, dass basierend auf der ersten Zahl bzw. dem letzten Zwischenergebnis ein Operator und eine neue Zahl eingegeben werden. Daraufhin wird die Rechnung sofort ausgeführt und das neue Zwischenergebnis angezeigt. Nach Drücken der Taste '=' wird das Endergebnis angezeigt und der Programmablauf endet.

- 6) Schreiben Sie ein Programm, das alle Zahlen von 0 bis 99 zweistellig ausgibt, wobei diejenigen Zahlen durch zwei Sterne (\*\*) ersetzt werden sollen, deren Quersumme 7 ist und die durch 7 teilbar sind (logisches UND)!

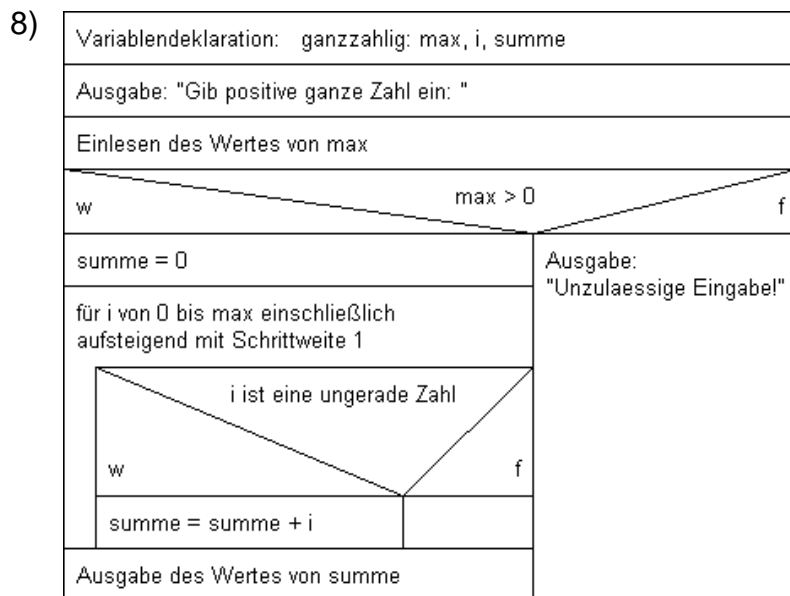
Sorgen Sie jeweils für eine übersichtliche Ausgabe mit je 10 Zahlen pro Zeile getrennt durch Leerzeichen!

- 7) Ein Programm soll einen eingelesenen ganzzahligen, nichtnegativen Sekundenbetrag in die Darstellung Wochen, Tage, Stunden, Minuten, Sekunden umwandeln und ausgeben. Beispiel: 6574548s = 10 Wochen, 6 Tage, 2:15:48 Stunden.

Beachten Sie:

Die Ausgaben „Woche“ bzw. „Wochen“ und „Tag“ bzw. „Tage“ sollen sprachlich korrekt erfolgen, also z. B. „1 Tag“, aber „3 Tage“ !

In einem Programmlauf soll mehr als ein Sekundenwert umgewandelt werden können! Überlegen Sie sich dazu ein geeignetes Abbruchkriterium.



- a) Was bewirkt nebenstehender Algorithmus, wenn der Benutzer die Zahl 8 eingibt?

Welche Ausgabe erfolgt, wenn der Benutzer die Zahl 0 eingibt? Arbeiten Sie den Algorithmus Schritt für Schritt ab und notieren Sie, wie sich die Werte der Variablen ändern!

- b) Formulieren Sie diesen Algorithmus als ein C-Programm!