

3. Übung: Funktionen

Ziel: Zerlegung einer Aufgabe in mehrere Funktionen
(noch kein objektorientierter Ansatz).
Weiteres Üben der Programmierung von Schleifen.

Aufgabe: Iterative Berechnung einer mathematischen Funktion

Gegeben ist folgender Algorithmus zur Berechnung der Quadratwurzel einer reellen Zahl:

Sei $a > 0.0$ eine reelle Zahl. Dann konvergiert die rekursiv definierte Folge

$$x_0 > 0 \quad \text{beliebiger Anfangswert}$$
$$x_{n+1} = \frac{1}{2} * (x_n + a/x_n)$$

gegen die Quadratwurzel von a .

Setzen Sie diese Definition in eine Schleife um, die bei jedem Durchlauf die nächst bessere Näherung berechnet und in einer Variablen ablegt (x_n ist dann der Wert der Variablen x im n -ten Schleifendurchlauf).
Entwickeln Sie nach diesem Prinzip eine Funktion zur Berechnung von Wurzeln mit einer wählbaren Genauigkeit ϵ . Sie dürfen dabei keine Funktion aus `<math.h>` einsetzen!
Schreiben Sie zu Testzwecken ein Hauptprogramm, das die Zahl a vom Benutzer abfragt und die Wurzel am Bildschirm ausgibt.

2. Aufgabe: Wiederverwendung von Code

Ist die Anzahl der Primzahlen, die kleiner als 10000 sind, selber eine Primzahl? Schreiben Sie ein Programm, das die Antwort berechnet und ausgibt. Verwenden Sie zu diesem Zweck Ihren Code aus der 3. Aufgabe der 2. Übung und "verpacken" Sie ihn in eine Funktion.

3. Aufgabe: Zerlegung in Funktionen

Ein Programm soll ein 2-dimensionales Array ganzer Zahlen zeilenweise von der Tastatur einlesen. Die Größe des Arrays ist variabel (aber maximal 10x10 Elemente; deklarieren Sie ein Array fester Größe, das gegebenenfalls nur teilweise genutzt wird) und wird zu Beginn vom Benutzer abgefragt. Die Elemente des Arrays sollen dann in einer "diagonalen Folge" ausgegeben werden.

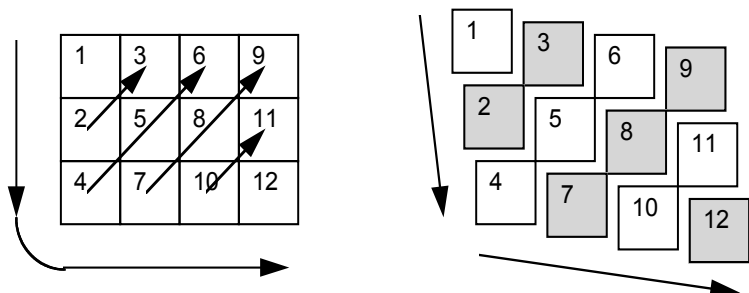
Beispiel:

Eingabereihenfolge:

1,3,6,9,2,5,8,11,4,7,10,12

Ausgabereihenfolge:

1,2,3,4,5,6,7,8,9,10,11,12



Strukturierung der Lösung:

Zerlegen Sie die

Gesamtfunktionalität in 3 Funktionen, die jeweils zuständig sind für:

- Einlesen der Größe und des Arrays selbst (fehlerhafte Eingabe der Größe abfangen !)
- Ausgabe einer "Diagonalen" des Arrays, beginnend an einem beliebigen Element, das durch seinen Zeilen- und Spaltenindex beschrieben wird
- Ausgabe des kompletten Arrays in "diagonaler Sortierung" unter Verwendung der Funktion b)