

Felder / Arrays

Wichtig: Erstellen Sie vor dem Programmieren jeweils ein Struktogramm!

- 1) Erstellen Sie ein Programm, das eine Folge von maximal 30 Zahlen vom Benutzer zur Eingabe anfordert und in einem Feld speichert.

Anschließend sollen die eingegebenen Zahlen in umgekehrter Reihenfolge wieder ausgegeben werden, wobei in jeder Ausgabezeile 5 Zahlen stehen sollen.

Dazu soll noch die kleinste und die größte Zahl bestimmt und ausgegeben werden!

Schreiben Sie das Programm so, dass der Benutzer in einem Programmlauf mehrere Zahlenfolgen nacheinander eingeben kann!

- 2) Erzeugen Sie in einem Programm zu einer Folge von maximal 20 Zahlen, die vom Benutzer einzugeben sind, das Zahlendreieck, in dem sich ab der zweiten Zeile jede Zahl als Summe der darüberliegenden Zahl und deren rechter Nachbarzahl ergibt.

Beispiel mit 4 Zahlen:

3	8	12	6
11	20	18	
31	38		
69			

Sie können die Aufgabe mit einem eindimensionalen Feld lösen, in dem die Ausgangszahlenfolge (1. Zeile) gespeichert wird und in dem sukzessive die nächsten Zeilen berechnet und ausgegeben werden. Sie können aber auch ein zweidimensionales Feld benutzen und die vollständige Berechnung vor der Ausgabe durchführen.

Schreiben Sie das Programm so, dass der Benutzer in einem Programmlauf mehrere Zahlendreiecke nacheinander berechnen lassen kann!

- 3a) Schreiben Sie ein Programm zur Berechnung der Summe und des Skalarproduktes von zwei Vektoren einer Dimension von maximal 20!

Die Komponenten eines Vektors sollen dabei in einem Array gespeichert werden!

In einer ersten Programmversion können Sie die Werte der Vektorkomponenten bei der Felddefinition festlegen und später erst die Eingabe der Vektorkomponenten über die Tastatur ermöglichen.

Sorgen sie für eine übersichtliche Ausgabe der Ergebnisse!

Hinweis:

Für die Vektoren $\vec{x} = (x_1, x_2, \dots, x_n)^T$ und $\vec{y} = (y_1, y_2, \dots, y_n)^T$ ist

$\vec{x} + \vec{y} = (x_1+y_1, x_2+y_2, \dots, x_n+y_n)^T$ die Summe der beiden Vektoren und

$\vec{x} \cdot \vec{y} = x_1 \cdot y_1 + x_2 \cdot y_2 + \dots + x_n \cdot y_n$ das Skalarprodukt der beiden Vektoren.

b) Lösen Sie diese Aufgabe mit Hilfe von C-Funktionen für folgende Teilaufgaben:

- Eingabe eines Vektors durch den Benutzer,
- Ausgabe eines Vektors,
- Berechnung des Summenvektors von zwei Vektoren,
- Berechnung des Skalarproduktes von zwei Vektoren.

Im Hauptprogramm soll die Dimension der Vektoren vom Benutzer erfragt werden und der Speicherplatz für die Vektoren dynamisch bereitgestellt werden!

4) Schreiben Sie in Anlehnung an das Beispiel

„5-1-Messwerte: Speichern von Messwerten, Berechnung von Kennwerten“

eine Funktion zur Berechnung der Kennwerte Minimum, Maximum und Median (das ist der mittlere Wert, genaue Definition siehe unten!) von Messwerten, die in einem Feld gespeichert sind. Sie können dabei auf die Funktion „sortieren“ aus dem Beispiel „Sortieren einer Zahlenfolge“ zurückgreifen.

Beachten Sie: Die Reihenfolge der Messwerte in dem Feld, in dem sie beim Einlesen abgelegt wurden, soll von dieser Funktion nicht verändert werden.

Das Hauptprogramm soll wie im Beispiel „5-1-Messwerte“ die Eingabe der Messwerte und die Ausgabe der Ergebnisse veranlassen.

Definition des Median/Zentralwert:

x_1, x_2, \dots, x_n seien n Messwerte und $x_{(1)}, x_{(2)}, \dots, x_{(n)}$ seien die aufsteigend sortierten Messwerte.

Der Median/Zentralwert \tilde{x} der n Messwerte ist definiert als

$x_{((n+1)/2)}$, falls n ungerade ist, und als $(x_{(n/2)} + x_{(n/2+1)})/2$, falls n gerade ist.

5a) Schreiben Sie ein Programm zur Multiplikation von zwei Matrizen!

Sie können die Werte der Matrizen bei der Felddefinition festlegen und erst in Teilaufgabe (b) die Eingabe der Matrizen über die Tastatur ermöglichen.

Sorgen sie für eine übersichtliche Ausgabe der Ergebnismatrix!

Hinweis:

Eine rechteckige Anordnung von $n \cdot m$ reellen Zahlen in m Zeilen und n Spalten nennt man in der Mathematik eine $(m \times n)$ -Matrix.

Die Zahl in der i -ten Zeile und j -ten Spalte der Matrix A wird als Matrixelement a_{ij} bezeichnet.

Wenn A eine $(m \times n)$ -Matrix mit den Elementen a_{ij} und B eine $(n \times r)$ -Matrix mit den Elementen b_{ij} ist, so ist die Matrix $C = A \cdot B$ eine $(m \times r)$ -Matrix mit den Elementen c_{ij} , wobei gilt:

$$c_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \dots + a_{in} \cdot b_{nj}$$

Das Element c_{ij} der Produktmatrix C ist also das Skalarprodukt aus der i -ten Zeile von A und der j -ten Spalte von B .

b) Lösen Sie diese Aufgabe mit Hilfe von C-Funktionen für folgende Teilaufgaben:

- dynamische Definition einer Matrix (siehe „5-5-3-matrix-ein-aus“),
- Eingabe einer Matrix durch den Benutzer (siehe „5-5-3-matrix-ein-aus“),
- Ausgabe einer Matrix (siehe „5-5-3-matrix-ein-aus“),
- Berechnung des Produktes von zwei Matrizen.

Im Hauptprogramm soll vom Benutzer die Zeilen- und Spaltenanzahl der Matrix erfragt werden.

6)

Variablendeklaration: Zeichen: weiter Ganzzahl: zahl, i, z[10] Gleitkommazahl: prod
Ausgabe: "Gib ganze Zahl >0 und <=20 ein: "
Einlesen des Wertes von zahl
solange zahl<1 oder zahl>20
Ausgabe eines Extrahinweises auf den Eingabefehler
Ausgabe: "Gib ganze Zahl >0 und <=20 ein: "
Einlesen des Wertes von zahl
für i von 0 bis 9 mit Schrittweite 1
z[i] = (i+1)*zahl
prod = 1
i = 1
solange i<=10
Ausgabe von i und z[i-1]
prod = prod*z[i-1]
i = i+1
Ausgabe von prod
Ausgabe: "Das Ganze nochmal?"
Löschen des Eingabepuffers und Einlesen des Wertes von weiter
solange weiter='j' oder weiter='J'
Ausgabe: "Programmende!"

- Was bewirkt nebenstehender Algorithmus, wenn der Benutzer die Zahl 3 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!
- Beschreiben Sie mit Worten, was der Algorithmus leistet!
- Implementieren Sie den Algorithmus als ein C-Programm!

- Erstellen Sie ein Programm, das 256 Ganzzahlwerte auf dynamisch allokiertem Speicherplatz abspeichert und wieder ausgibt.

Gehen sie dabei im Einzelnen so vor:

- Erzeugen Sie dynamisch Speicherplatz für 256 `int`-Werte!
Die Startadresse dieses Speicherplatzes soll `addr_var` heißen!
- Geben Sie `addr_var` und `&addr_var` jeweils in hexadezimaler Schreibweise aus!
Wie groß ist der Speicherplatzbedarf von `addr_var` ?
- Speichern Sie auf dem allokierten Speicherplatz die Werte von 0 bis 255 ab und geben Sie zur Kontrolle diese Werte mit 8 Zahlen pro Zeile übersichtlich aus!
- Was passiert, wenn Sie (a) in einem Programm mehrfach nacheinander mit dem gleichen Namen für die Startadresse ausführen? Ist dieses Vorgehen sinnvoll?
- Was passiert, wenn Sie in (a) Speicherplatz für eine Milliarde ($=10^9$) `int`-Werte allokieren wollen?