

### Aufgabe 1 (Funktionen, Beispiel Quader):

Es sollen der Rauminhalt und die Oberfläche eines Quaders berechnet und am Bildschirm angezeigt werden. Dazu sind normalerweise 3 Parameter notwendig: Höhe, Breite und Tiefe. Da der Quader beliebig im dreidimensionalen Raum gedreht werden kann, sind die drei Abmessungen gleichwertig, Sie können sie auch Seite1, Seite2 und Seite3 nennen... Alle Abmessungen sind in cm anzugeben.

1. Schreiben Sie dazu eine Funktion, die sowohl mit einem, mit zwei, mit allen drei Parametern oder ganz ohne Parameter aufgerufen werden kann und die fehlenden Angaben nachträglich vom Benutzer erfragt, damit alle 3 Parameter zur Berechnung zur Verfügung stehen. Testen Sie Ihre Funktion im Hauptprogramm, indem Sie die Funktion auf die 4 möglichen Arten aufrufen. Sie können die Werte für die Abmessungen fest programmiert vorgeben (Benutzereingabe sei nicht erforderlich, können Sie aber vorsehen, wenn Sie möchten).
2. Schreiben Sie eine weitere Funktion, die – wenn sie nicht mit allen Parametern aufgerufen wird – für die fehlenden Seitenlängen einen Standardwert von 10 cm annimmt. Testen Sie auch diese Funktion durch 4 verschiedene Aufrufe im Hauptprogramm.
3. Sehen Sie nun eine weitere Funktion vor, die von Ihren beiden bisherigen Funktionen aufgerufen wird, um (wenn alle 3 Parameter vorliegen) die benötigten Berechnungen und Ausgaben vorzunehmen. Berechnung des Quaders und Ergebnisanzeige sollen also nur ein einziges Mal in Ihrem Gesamtprogramm vorkommen!
4. Implementieren Sie nun eine Möglichkeit, die zum BEGINN Ihres Programms vom Benutzer abfragt, ob die Quader-Berechnung bei fehlenden Angaben mit den Default-Werten ODER mit den nachträglich abzufragenden Abmessungen durchgeführt werden soll. Je nach dem soll dann auch in Ihrem Test-Szenario nur die eine oder die andere Variante ausgeführt werden.

Achten Sie GENERELL einmal darauf, dass Sie sämtliche Aktionen, die während des Programmablaufes mehrmals auftreten können (zum Beispiel die Anfrage an den Benutzer, die fehlende 3. Kantenlänge anzugeben) nur ein Mal ausprogrammieren und durch geschickte Verwendung der Funktionen mehrmals verwenden können!

```
##### Programm zur Berechnung von Quadern #####
Soll bei fehlenden Abmessungen eine Standardlaenge von 10 cm angenommen werden?
(Druecken Sie 1 fuer Ja, 0 oder eine andere Zahl fuer Nein) 1
Berechnung eines Quaders mit den Kantenlaengen:
    Seite 1: 10 cm
    Seite 2: 10 cm
    Seite 3: 10 cm
    Rauminhalt : 1000 ccm
    Oberflaeche: 600 ccm
Berechnung eines Quaders mit den Kantenlaengen:
    Seite 1: 1.23 cm
    Seite 2: 10 cm
    Seite 3: 10 cm
    Rauminhalt : 123 ccm
    Oberflaeche: 249.2 ccm
Berechnung eines Quaders mit den Kantenlaengen:
    Seite 1: 1.23 cm
    Seite 2: 4.56 cm
    Seite 3: 10 cm
    Rauminhalt : 56.088 ccm
    Oberflaeche: 127.018 ccm
Berechnung eines Quaders mit den Kantenlaengen:
    Seite 1: 1.23 cm
    Seite 2: 4.56 cm
    Seite 3: 7.89 cm
    Rauminhalt : 44.2534 ccm
    Oberflaeche: 102.584 ccm
Druecken Sie eine beliebige Taste . . .
```

## Aufgabe 2 (Datenfeld mit Zeigern):

Deklariieren Sie 2 Datenfelder vom Typ bool und float mit jeweils 5 Elementen und initialisieren sie mit passenden Werten.

1. Lassen Sie sich anzeigen, wo im RAM diese Datenfelder gespeichert werden.
2. Verändern Sie den Wert des jeweils dritten Elements der Datenfelder mit Hilfe eines Zeigers. Verwenden Sie für jedes Datenfeld einen eigenen Zeiger.
3. Zeigen Sie die veränderten Werte an diesen Adressen an, indem Sie nun wieder die Datenfeldnamen mit Index verwenden.

## Aufgabe 3 (Dynamisches Array erzeugen):

Fordern Sie weiteren Arbeitsspeicher für Datenfelder (Arrays) an:

Der Benutzer möge angeben, wie viele Elemente die Datenfelder enthalten sollen. Der Benutzer soll Datenfelder mit mindestens 10 Elementen wählen.

### 1. Erzeugen Sie nun zunächst ein Datenfeld für die gewünschte Anzahl von bool-Werten.

Lassen Sie anzeigen, an welcher Stelle im RAM Ihnen neuer Arbeitsspeicher zugewiesen wurde. Zeigen Sie die (jetzt ja noch nicht initialisierten) Werte des neuen Datenfeldes an.

Kopieren Sie nun wiederholt die Werte des aus Aufgabe 2 bekannten bool-Datenfeldes in das neu erstellte Datenfeld. Es werden also kopiert:

- altes Datenfeld – Element-Index: 0 1 2 3 4 0 1 2 3 4 0 1 2 3 ...
- neues Datenfeld- Element-Index: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 ...

Zeigen Sie zur Kontrolle die nun gültigen Werte des neuen Datenfeldes an.

**Tipp:** Verwenden Sie für die Berechnung der Index-Werte des alten Datenfeldes die Modulo-Division!

**Tipp:** Um Schreibarbeit zu sparen, dürfen Sie die Anzeige der nicht-initialisierten Werte, die Kopier-(Zuweisungs-)Anweisung und die Anzeige der neuen Werte in einer einzigen Schleife einbauen. Eine Beispiel-Ausgabe führe ich Ihnen am Beamer während des Praktikums vor.

Geben Sie den angeforderten Speicher NICHT frei (kein „delete“).

### 2. Erzeugen Sie nun ein Datenfeld für die gleiche Anzahl von float-Werten.

Zeigen Sie sinngemäß wie in obiger Aufgabe die zugewiesene Adresse und die nicht-initialisierten Werte des neuen Datenfeldes an. Kopieren Sie ebenfalls sinngemäß die Werte des alten Datenfeldes in das neue float-Array und geben die neuen Werte am Bildschirm aus.

Geben Sie den soeben für das float-Array angeforderten Speicherplatz wieder frei.

Erzeugen Sie abermals ein neues float-Datenfeld und geben ihm einen neuen Namen. Wiederholen Sie die beschriebene Prozedur (Anzeigen der alten Werte, Kopieren und Anzeige der neuen Werte) für dieses Datenfeld. Was stellen Sie fest?

Geben Sie nun den Speicherplatz des bool-Datenfeldes aus dieser Aufgabe frei und erzeugen ein weiteres float-Datenfeld wie hier mehrmals beschrieben... Was stellen Sie fest?

Erzeugen Sie nun wieder ein neues bool-Datenfeld mit dem gleichen „Spielchen“... Was stellen Sie fest?

Geben Sie allen angeforderten Speicher wieder frei und beenden das Programm.

Starten Sie das Programm mehrmals und wählen jeweils eine andere Anzahl von Elementen für das neue Datenfeld. Was stellen Sie fest?