

Praktikum Programmieren  
Übungsblatt 8  
Termin: 7. Dezember 2023

### Übung 1 „Strukturen“

Definieren Sie eine Struktur `Punkt` mit Feldern `x` und `y` – jeweils vom Typ `int`. Führen Sie zwei Variable `p1` und `p2` ein und initialisieren Sie die `x`- bzw. `y`-Werte mit Zahlen Ihrer Wahl.

Berechnen Sie die Summe der `x`- und `y`-Werte und weisen Sie die Werte einer dritten Variablen, auch vom Typ `Punkt`, zu.

### Übung 2 „Arrays und Strukturen“

Erstellen Sie ein Array der Länge 50 vom Typ `Punkt` aus Aufgabe 1 und füllen Sie dieses. Die jeweiligen `x`- und `y`-Werte können mit Zufallszahlen belegt werden (siehe Aufgabenblatt 1, Funktion `rand()`).

Iterieren Sie nachfolgend über das Array und finden Sie die Position der Punkte mit dem kleinsten `x`- bzw. `y`-Wert.

### Übung 3 „Datum“

Definieren Sie eine Struktur `Datum` mit den Attributen `tag`, `monat` und `jahr`. Der Tag und das Jahr sind jeweils vom Typ `int`, der Monat soll als Aufzählungstyp (`enum`) modelliert werden.

Deklarieren Sie eine Variable vom Typ `Datum` und setzen Sie das Datum auf den 7. Dezember 2023.

Implementieren und testen Sie die Funktion `druckeDatum`, die ein Datum ausgeben, in den folgenden Varianten:

- `void druckeDatum(Datum d)`
- `void druckeDatum(Datum &d)`
- `void druckeDatum(Datum *d)`

Machen Sie sich die Unterschiede bei der Parameterübergabe bewusst.

### Übung 4 „Funktionen mit Strukturen als Parameter“

Schreiben Sie eine Funktion `bool bevor(const Datum& a, const Datum& b)`, die genau dann den Wert `true` zurückgibt, wenn das Datum `a` vor dem Datum `b` liegt. Nutzen Sie dazu die Struktur `Datum` aus Aufgabe 3.

Testen Sie die Funktion mit mindestens den folgenden Fällen:

- Zwei gleiche Daten
- Zwei Daten mit gleichem Jahr und Monat aber unterschiedlichen Tagen
- Zwei Daten mit gleichem Jahr aber unterschiedlichen Monaten und Tagen
- Zwei Daten mit unterschiedlichen Jahren

Vertauschen Sie bei unterschiedlichen Jahren die Reihenfolge der Jahre jeweils.

## Übung 5 „Vektorfunktionen“

Definieren Sie eine Struktur **Vektor** mit Feldern **x** und **y** – jeweils vom Typ **float**.

Implementieren Sie die folgenden Funktionen:

- **float skalarProdukt(const Vektor& a, const Vektor& b)**: liefert den Wert des Skalarprodukts der Vektoren **a** und **b**
- **Vektor vektorSumme(const Vektor& a, const Vektor& b)**: liefert die Summe der Vektoren **a** und **b**
- **Vektor kreuzProdukt(const Vektor& a, const Vektor& b)**: liefert das Kreuzprodukt der Vektoren **a** und **b**
- **Vektor skalierterVektor(float faktor, const Vektor& x)**: liefert das Ergebnis der Multiplikation des Skalars **faktor** mit dem Vektor **x**.

Deklarieren und initialisieren Sie verschiedene Variablen vom Typ **Vektor** und testen Sie Ihre Funktionen damit.

## Übung 6 „Personendatenbank“

Definieren Sie eine Struktur **Person** mit Feldern **name**, **vorname** und **geburtsdatum**. Verwenden Sie für **name** und **vorname** jeweils ein Array mit Elementtyp **char** und ausreichender Länge, mindestens jedoch jeweils 20 Zeichen. Für das Feld **geburtsdatum** verwenden Sie die Struktur **Datum** aus Aufgabe 3.

Erstellen Sie ein Array mit Elementtyp **Person** und initialisieren Sie es mit Daten von mindestens 5 Personen der Weltgeschichte (z.B. Isaac Newton, Gottfried Leibniz, Albert Einstein). Erstellen Sie außerdem eine Variable **personenzahl** vom Typ **int** und initialisieren Sie sie mit der Zahl von Einträgen in Ihrem Array.

Schreiben Sie eine Funktion **findeGeburtsdatum**, die den einen Namen und einen Vornamen übergeben erhält, den Eintrag mit dem entsprechenden Namen und Vornamen in der Liste findet und den Namen sowie das Geburtsdatum ausgibt („Albert Einstein wurde am 14. März 1879 geboren“).

Wenn kein Eintrag mit dem entsprechenden Namen existiert, geben Sie eine entsprechende Fehlermeldung aus.

Verwenden Sie für den Vergleich der Strings die Funktion **strcmp** aus der Bibliothek **<string.h>**.

## Übung 7 „Geschachtelte Schleifen“

Betrachten Sie das folgende Programm:

```
1 #include <stdio.h>
2
3 int main(){
4     int zahlen[5];
5     int i, j;
6     for (i = 20; i > 15; i=i-1){
7         zahlen[20-i] = i % 3;
8     }
```

```
9  
10  for (i = 0; i < 5; i=i+1){  
11      for (j = 5; j > 0; j=j-2){  
12          zahlen[5-j] = zahlen[i/2] + zahlen[j-1];  
13      }  
14  }  
15  return 0;  
16 }
```

Welche Werte enthält das Array **zahlen** jeweils vor Eintritt in die innere der geschachtelten **for**-Schleife?

Überlegen Sie sich die Lösung zunächst auf Papier und „fragen“ Sie danach Ihren Rechner.