



Übung 7

Aufgabe 1: Reelle Zahlen - Winkel und Skalarprodukt

Sie haben mittlerweile gelernt, wie man

- Daten besser strukturiert,
- Funktionen und Prozeduren schreibt.

Wenden Sie dieses Wissen in dieser Aufgabe auf die Berechnung von Winkel und Skalarprodukt aus Übung 1, Aufgabe 2 an.

Aufgabe:

Formulieren Sie das Programm aus Übung 1, Aufgabe 2 neu unter folgenden Gesichtspunkten:

- Entwerfen Sie eine geeignete Datenstruktur für Vektoren.
- Entwerfen Sie je eine Funktion/Prozedur zur Berechnung des Skalarproduktes und des Winkels.
- Entwerfen Sie eine Funktion/Prozedur zur Eingabe der Vektoren.

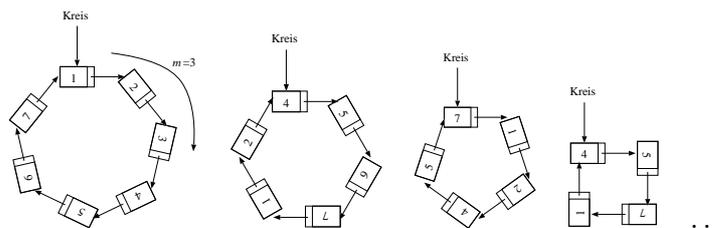
Aufgabe 2: Kinderabzählreime

In dieser Aufgabe geht es darum, die Verwendung von Pointern und verketteten Listen anhand eines kleinen Beispiels zu üben:

In einem Kreis stehen n Kinder und spielen folgendes Spiel: Sie benutzen einen Abzählreimspiel, der immer um m Personen weiterzählt. Jeweils das letzte Kind muss ausscheiden, das vom Abzählreim getroffen wird. Der Kreis schliesst sich und es wird beim darauffolgenden Kind von neuem abgezählt. Gewonnen hat dasjenige Kind, das am Schluss übrig bleibt.

Simulation am Computer: Wir wollen dieses Kinderabzählreim mit zyklisch verketteten Listen am Computer simulieren. Es soll ein Programm geschrieben werden, das die Reihenfolge auflistet, in der die Kinder ausscheiden. Das Programm bekommt als Eingabe die Zahl n der Kinder am Anfang im Kreis und die Zahl m , um die weitergezählt wird, bis wieder jemand ausscheiden muss.

Die folgende Grafik zeigt am Beispiel $n = 7, m = 3$, wie sich die Datenstruktur entwickeln soll:



Ausgabe: 3 6 2 7 5 1 Gewinner: 4

Ausgabe:

- a) Entwerfen Sie eine Datenstruktur für ein Kind (wir nennen sie `Kind`) und den Kreis (wir nennen diese Variable `Kreis`).
- b) Schreiben Sie eine Prozedur,


```
void anfang(Kind *&Kreis, int n)
```

 die die Datenstruktur zu anfang aufbaut: Die Zeiger `Kreis` zeigt nach der Ausführung der Prozedur auf die erste der n Personen im Kreis.
- c) Schreiben Sie ein Prozedur


```
void abzaehltreim(Kind *&Kreis, int m)
```

 die die Liste in der beschriebenen Art abbaut: Es soll solange jedes m -te Kind aus der Liste `Kreis` entfernt werden, bis nur noch eines übrig ist. Jedesmal, wenn ein Kind aus der Liste entfernt wird, wird dessen Nummer ausgegeben.
- d) Vervollständigen Sie alle Teile zu einem lauffähigen Programm, das nach der Eingabe von n und m die Reihenfolge der ausscheidenden Kinder (sowie den Gewinner) ausgibt.