

Übungsblatt 6: Programmieren in C (WS 2019/20)

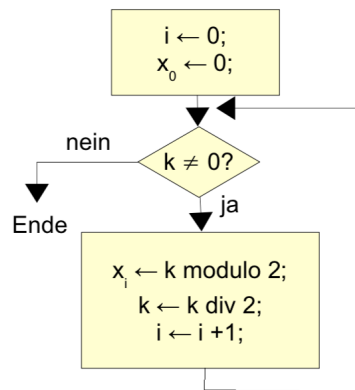
Abgabe: Montag, 16.12.19, 12:00

Aufgabe 1 Zahlenkonvertierung I (10 Punkte)

Abgabe: septenaer.c, flussdiagramm.pdf

Im Land der 7 Zwerge hinter den 7 Bergen wird das Septenärsystem verwendet, d.h. ein Stellenwertsystem mit der Basis 7. Es verwendet die Ziffern 0, 1, 2, 3, 4, 5 und 6, um Zahlen darzustellen.

- In der Vorlesung wurde folgender Algorithmus zur Umrechnung von Dezimal- in Dualzahlen vorgestellt.



Wandeln Sie diesen Algorithmus für die Umrechnung von Dezimal- in Zahlen zur Basis 7 ab und erweitern Sie ihn um die Eingabe einer Dezimalzahl und die Ausgabe. Beschreiben Sie den Algorithmus in Form eines Flussdiagramms (Abgabe `flussdiagramm.pdf`).

- Implementieren Sie den Algorithmus (Abgabe `septenaer.c`). Die Ausgabe soll in Tabellenform analog zu der Tabelle auf den Vorlesungsfolien erfolgen. Das heißt, bei der Eingabe der Zahl 777 soll die Bildschirmausgabe wie folgt aussehen:

k	k modulo 7	k div 7	i
777	0	111	0
111	6	15	1
15	1	2	2
2	2	0	3
0	-> Ende		

Für die Ausrichtung der Tabelleneinträge verwenden Sie die formatierte Ausgabe von Integern mittels `printf()`.

Aufgabe 2 Zahlenkonvertierung II (5 Punkte)

Abgabe: `from_septenaer.c`

Schreiben Sie ein Programm, das eine positive ganze Zahl in Darstellung zur Basis 7 als Eingabe nimmt und die entsprechende Dezimalzahl ausgibt.

Beispiel: Bei der Eingabe von 51 soll die Ausgabe 36 sein.

Aufgabe 3 *Aufsteigende Folge von Zahlen (5 Punkte)*

Abgabe: `aufsteigend.c`

Schreiben Sie ein Programm, das als erstes eine positive ganze Zahl `n` einliest, dann `n` Integer einliest und als Ergebnis `aufsteigend` ausgibt, falls die `n` Zahlen aufsteigend geordnet sind, sonst `nicht aufsteigend`.

Beispiel: Bei der Eingabe von

5 -10 2 6 6 29

soll die Ausgabe `aufsteigend` sein.

Bei der Eingabe von

4 1 0 6 8

soll die Ausgabe `nicht aufsteigend` sein.

Aufgabe 4 *Zugriff auf Komponenten von Strukturen (5 Punkte)*

Abgabe: `zeit.c`

Übernehmen Sie das Beispiel zur Systemzeit in C/Unix von den Vorlesungsfolien (Kap. 03, Seite 96), übersetzen es und führen Sie es aus. Ändern Sie das Programm so ab, dass es die Sekundenanzahl von der Konsole einliest und ändern Sie die Ausgabe, so dass bei Eingabe von 1543165768 folgendes ausgegeben wird:

Sonntag, den 25. November 2018, 18 Uhr 9

Hinweis: Verwenden Sie zum Umwandeln von Eingabe folgende Anweisungen:

```
scanf("%ld", &sekunden);           // Einlesen der Sekundenanzahl
ortszeit = localtime(&sekunden);   // Ermitteln der Ortszeit
```

Aufgabe 5 *Arrays von Strukturen (15 Punkte)*

Abgabe: `exercisegroup.c`

Für die Verwaltung der Übungsaufgaben benötigen wir eine Tabelle mit den Namen und Punkteständen der Studierenden. Die Übungsgruppentabelle soll dabei maximal 10 Studierenden aufnehmen können. Schreiben Sie ein C-Programm, das die folgenden Schritte ausführt:

- Definieren Sie zunächst einen Strukturtyp `studi`, der den Namen und den Punktestand eines Studierenden speichern kann. Ein Studierendenname soll nur aus einem Wort bestehen (max. 30 Zeichen); der Punktestand ist eine natürliche ganze Zahl.
- Lesen Sie zunächst 10 verschiedene Studierendendaten von der Konsole ein und speichern Sie diese in einem Array ab.
- Geben Sie dann die Daten in der eingegebenen Reihenfolge wieder aus. Definieren Sie dazu einen Pointer auf Strukturen vom Typ `studi`. Iterieren Sie dann über die Einträge des Arrays und geben Sie jeweils Name und Punktestand aus.
- Geben Sie dann die Studierendendaten in umgekehrter Reihenfolge aus.
- Ermitteln Sie die Summe der Punkte aller Studierenden.
- Ermitteln Sie den Studierenden mit dem höchsten Punktestand.

Beispiel

Eingabe:

```
10, anna
16, bert
19, carl
25, david
11, erwin
78, franz
56, gustav
98, hans
20, immi
91, jan
```

Ausgabe:

```
anna : 10
bert : 16
carl : 19
david : 25
erwin : 11
franz : 78
gustav : 56
hans : 98
immi : 20
jan : 91
jan : 91
immi : 20
hans : 98
gustav : 56
franz : 78
erwin : 11
david : 25
carl : 19
bert : 16
anna : 10
Gesamtpunktzahl: 424
Bester Spieler ist hans mit 98 Punkt(en).
```

Hinweis: Zum Einlesen **einer** Zeile verwenden Sie (Abwandlungen) folgende Anweisungen:

```
// Benötigt die String- und IO-Bibliothek
#include <stdio.h>
#include <string.h>

...
int punkte;
char name[MAX_LAENGE_NAME+1];

// Einlesen der Punkte
scanf("%d, ", &punkte);

// Einlesen von max. MAX_LAENGE_NAME vielen Zeichen
fgets(name, MAX_LAENGE_NAME, stdin);

// Entfernen des Zeilenumbruchs am Ende der Eingabe
int j = strcspn(name, "\n");
name[j] = '\0';
```