

Arbeitsblatt: INF1

Name:

Kurznamen:

Strings, Funktionen, Datenstrukturen

Aufgabe 1: String Funktionen

Beim Programmieren in C benötigt man regelmässig Funktionen aus der C-Standardbibliothek, zum Beispiel *printf(...)* aus der *stdio.h*, *floor(...)* aus der *math.h* oder *system(...)* aus der *stdlib.h*.

Machen Sie sich mit der String Bibliothek vertraut:

1. Lesen Sie einen String (Namen) ein und geben Sie ihn wieder aus
 - mit Formatstring "%s" und "%19[^\n]s".
 - überlegen Sie wieso, welches die Vorteile der zweiten Variante sind
2. Fügen Sie diesen Namen dem String "Hello " hinzu.
3. Geben Sie den String als Ganzes wieder aus: e.g. "Hello Hugo"
4. Geben Sie zusätzlich die Länge des Strings aus
5. Prüfen Sie ob der Namen "Hugo" heisst und
 - geben Sie "Hello Hugo" aus, falls das zutrifft
 - geben Sie "Hello Unbekannt" aus, falls das nicht zutrifft
6. Ersetzen Sie im Eingabestring alle "u" und "o" durch "a"
7. Lesen Sie eine Ganz- und Fließkommazahl ein und geben Sie diese wieder aus.

Aufgabe 2: Zahlen sortieren

Ein nicht besonders effizientes, aber einfach zu implementierendes Sortiervorgehen ist der sog. Bubblesort. Vereinfachte Beschreibung: Gehe durch das Array und vertausche zwei benachbarte Elemente, wenn sie noch nicht in der richtigen Reihenfolge sind. Wiederhole dies, bis keine Vertauschung mehr stattfindet. Weitere Informationen zu diesem Sortiervorgehen können Sie dem Wikipedia-Artikel zu Bubblesort entnehmen (<http://de.wikipedia.org/wiki/Bubblesort>). Dort finden Sie auch den Algorithmus im Pseudocode:

```
bubbleSort(Array A) {  
    for (n=SIZE; n>1; n=n-1){  
        for (i=0; i<n-1; i=i+1){  
            if (A[i] > A[i+1]){  
                swap(&A[i], &A[i+1])  
            } // ende if  
        } // ende innere for-Schleife  
    } // ende äußere for-Schleife  
}
```

Implementieren Sie das Sortieren einer Zahlenfolge in C. Geben Sie die Zahlenfolge sortiert aus. Nehmen Sie folgendes Zahlenbeispiel

```
#define SIZE 5  
int a[SIZE] = {7, 12, 6, 43, 6};
```

Aufgabe: Schreiben Sie eine Funktion `void bubbleSort(int length, int a[])` mit der Sie eine Zahlenfolge sortieren können. Verwenden Sie die swap Funktion aus dem Script.

Abgabe

Praktikum: INF9.2

Filename: sortieren.c

Aufgabe 3: Rangliste erstellen

Die Zeitmessungen vom letzten mal möchten wir in eine Rangliste für einen Wettkampf erweitern. Dafür muss zuerst die Datenstruktur noch um den Namen ergänzt werden. In der `leseTeilnehmer` und `schreibeTeilnehmer` Funktionen soll der Namen ebenfalls eingelesen bzw. ausgegeben werden.

```
typedef struct {
    char name[20];
    int h, min, sec;
} Teilnehmer;

void teilnehmerAusgeben(int maxT, Teilnehmer* zeiten) ...
Teilnehmer leseTeilnehmer (void) ...
void teilnehmerSortieren(int size, Teilnehmer a[]) ...
```

Für die Rangliste sollen die Teilnehmer nach der Zeit sortiert werden. Dafür müssen die Zeiten miteinander verglichen werden können. Dafür verwenden wir eine Vergleichsfunktion `int vergleiche(Teilnehmer t1, Teilnehmer t2)`. Diese ist wie folgt definiert:

```
t1 < t2    → Wert kleiner 0
t1 = t2    → Wert: 0
t1 > t2    → Wert grösser 0.
```

Statt dem "<" in der Sortierungsfunktion wird nun die *vergleiche*-Funktion aufgerufen.

Hinweis:

- Verwenden Sie folgen Formatstring `scanf(" %19[^\n]s", t.name)` ; um den Namen einzulesen; sie erlaubt auch Blanks und stoppt automatisch nach 19 Zeichen.

Abgabe

Praktikum: INF9.3

Filename: rangliste.c

Aufgabe 4: Sortieren mit Funktionsparameter

Die Rangliste soll wieder nach der Zeit sortiert werden. Aber statt die vergleiche-Funktion direkt aufzurufen, möchten wir sie als Funktionsparameter übergeben. Das hat den Vorteil, dass der Sortierfunktion ein Sortierkriterium mitgegeben werden kann. Man könnte z.B. die Teilnehmer so einfach auch nach den Namen sortieren (optionale Aufgabe).

Aufgabe:

Schreiben Sie die Sortierfunktion so um, dass ein Funktionsparameter zum Vergleich mitgegeben werden kann. Sie können so die bestehende die bestehende vergleiche-Funktion übergeben.

```
typedef int (*Comparator)(Teilnehmer, Teilnehmer);  
void teilnehmerSortieren(int size, Teilnehmer a[], Comparator comp) {
```

Abgabe

Praktikum: INF9.4

Filename: rangliste.c