

Name : _____	Name : _____
Vorname : _____	Vorname : _____
Matrikel-Nr : _____	Matrikel-Nr : _____

**Ziele der Aufgabenstellung:**

- Programm mittels strukturierter Programmierung entwerfen
- Aufteilung des Programms in Module
- Einbindung eines Assemblerprogramms in ein C-Programm

**Vorzubereiten:**

- strukturierte Programmierung
- Aufteilung der Unterprogramme in sinnvolle Module
- Umgang mit Feldern und Zeigern, Call-by-reference-Aufrufkonvention
- Sortieren von Feldern

**Vorausgesetztes Wissen u. Praktikumstest:**

Anm.: Diese Themen werden Bestandteil des Praktikumstests sein.

- strukturierte Anweisungen in C  
(while, if..then..else, do...while, for(...), switch)
- definieren von Funktionen in C (call-by-value, call-by-reference)
- Umgang mit Strings, Feldern und Zeigern

**Aufgabenstellung:**

Gegeben sei ein Feld von Strings (*Stringliste*) sowie ein Feld von Zeigern darauf , z.B.

```
char *pMeineStrings[] = {  
    "Haller          25 EUR",  
    "Kandinsky       13 EUR",  
    "Brombach        5 EUR",  
    "Zaluskowski    120 EUR",  
    "Osman           17 EUR",  
    "\\0\\0"  
};
```

**Anm.:** Das Ende der Stringliste ist am String "\\0\\0" zu erkennen.

a) **Realisierung als reines C-Programm**

Zu schreiben ist ein Programm, welches die Strings entsprechend der Größe der Zahl im String sortiert und ausdrückt, also hier:

```
Brombach      5 EUR
Kandinsky     13 EUR
Osman         17 EUR
Haller        25 EUR
Zaluskowski  120 EUR
```

Folgende C-Unterprogramme sind zu erstellen:

```
/* Ausdrucken der Stringliste */
void PrintStringliste( ..... );

/* Sortieren einer Stringliste nach der Größe */
/* der Zahl im String. */
void SortiereStrings( ..... );

/* Zahl im String bestimmen und zurückgeben */
int  getNum( ..... );
```

Für die Sortierung kann das Sortierverfahren "Bubblesort" verwendet werden.

Die Ausgabe der Strings erfolgt mit *printf*. Als Terminal wird auf dem PC die Anwendung **terminal.bat** (Ordner LPC-Terminal) gestartet.

Die C-Unterprogramme (s.o.) sollen im Modul `stringsort.c` definiert sein.

Das startende Hauptprogramm (main) soll im Modul `main.c` definiert sein.

Die Funktionsdeklarationen sollen über eine Include-Datei (`stringsort.h`) bekannt gemacht werden.

b) **Realisierung als gemischtes C-/Assemblerprogramm**

Die C-Funktion `getNum()` soll jetzt ersetzt werden durch ein Assembler-Unterprogramm `getNum_asm`.

Hierfür wird dem Projekt eine Datei `getnum.s` zugefügt. Die Funktionsdeklaration für `getNum_asm` muss ebenfalls in `stringsort.h` eingetragen werden.

Das Programm (auch der Assemblerteil) soll

- vernünftig strukturiert und eingerückt
- sinnvoll kommentiert und
- mit selbsterklärenden Variablennamen versehen sein.

**Aufbau der Datei** getnum.s:

---

```
.global    getNum_asm

.align

# -----
# Inputregister  r0 (Stringadresse)
# Outputregister r0 (Zahlenwert)
# Verwendete Register retten und restaurieren.
# -----
getNum_asm:
    # Code starts here .....

    bx    lr

# -----

# GANZ WICHTIG !!!
.end
```

---

**Aufbau der Datei** main.c:

---

```
#include <stdio.h>
#include "stringsort.h"

// .... sonstige Headerdateien ....

char *pMeineStrings[] = {
    // die Strings (s.o.) ...
}

int main(void) {
    BaseStickConfig();

    // SortiereStrings() ...

    // PrintStringliste() ...

    while(1){} // Endlosschleife
}
```

---