

Aufgabenblatt 1

letzte Aktualisierung: 05. November, 20:06 Uhr

(62c87a8f4b7189691900186a5548e45ca9673fc6)

Ausgabe: Mittwoch, 5.11.2014

Abgabe: spätestens Freitag, 14.11.2014, 16:00

Thema: Finde kleinstes Element, Insertion Sort, Pseudocode

Abgabemodalitäten

- Alle abzugebenden Quelltexte müssen ohne Warnungen und Fehler auf den Rechnern des tubIT/IRB mittels `gcc -std=c99 -Wall` kompilieren.
- Die Abgabe erfolgt ausschließlich über SVN. Die finale Abgabe
 - Für Einzelabgaben erfolgt im Unterordner
Tutorien/t<xx>/Studierende/<tuBIT-Login>/Blatt<xx>/submission/
 - Für Gruppenabgaben erfolgt im Unterordner
Tutorien/t<xx>/Gruppen/g<xx>/Blatt<xx>/submission/
- Benutzen Sie für alle Abgaben von Programmcode das folgende Namensschema: `introprog.blatt0X.aufgabe0Y.Z.c`, wobei X durch die Blattnummer, Y durch die Aufgabe und Z durch die Unteraufgabe ersetzen ist.
Beispiel: `introprog.blatt01.aufgabe01.1.2.c`
Für jede Unteraufgabe geben Sie maximal eine Quellcodedatei ab, es sei denn, die Aufgabenstellung erfordert explizit die Abgabe mehrerer Dateien pro Aufgabe.
Alle anderen Abgaben (Pseudocode, Textaufgaben) benennen Sie wie oben beschrieben. Die zugelassenen Abgabeformate sind PDF, ODT und Text (txt). Auch hier verwenden Sie eine Datei pro Aufgabe.

Wie auch im C-Kurs werden Sie die Abgaben für die Vorlesung "Einführung in die Programmierung" im SVN abgeben. Dazu müssen Sie einmal ein neues SVN auschecken. Wechseln Sie mit dem Kommando `cd` in das Verzeichnis, wo das neue Repository liegen soll und geben dann folgenden Befehl ein:

```
svn co https://teaching.inet.tu-berlin.de/tubitauth/svn/introprog-ws1415
```

Nach einem erfolgreichen Checkout finden Sie in diesem Repository alle Materialien aus dem C-Kurs im Ordner C-Kurs, sowie eine Verzeichnisstruktur für die Vorlesung.

Wichtig: Ab Donnerstag, 13. November 2014 legen wir für Sie im SVN einen Ordner für Ihre entsprechende Gruppe an. Um diese und andere Änderungen unsererseits zu sehen, wechseln Sie in das oberste Verzeichnis des Repositoies und führen Sie das Kommando `svn up` aus. *Hinweis:* Das Kommando `svn up` aktualisiert die Inhalte im aktuellen Verzeichnis inklusive aller Unterverzeichnisse, übergeordnete Verzeichnisse werden NICHT aktualisiert. Daher empfehlen wir, `svn up` immer im obersten Verzeichnis des Repositories auszuführen.

```
-- Aufgaben
|  |-- Blatt01
|      |-- Blatt01.pdf
|      |-- vorgaben
|          |-- data
|              |-- unsorted_datafile.txt
|              |-- src
|                  |-- blatt01_aufgabe01_vorgabe.c
-- Material
|  |-- C-Kurs
|      |-- Vorlesungsfolien
|          |-- progintro-2014-00-orga.pdf
|          |-- ...
|  |-- IntroProg
|      |-- Vorlesungsfolien
-- Tutorien
|  |-- README.txt
|  |-- t01
|      |-- Gruppen
|          |-- g01
|              |-- Blatt01
|                  |-- feedback
|                  |-- logs
|                  |-- submission
|                  |-- workdir
|                  |-- info
|  |-- Material
|  |-- Studierende
|      |-- tubit23
|          |-- Blatt01
|              |-- feedback
|              |-- logs
|              |-- submission
|              |-- workdir
|              |-- info
```

Diese Grafik zeigt die Verzeichnisstruktur, für den tuBIT Nutzer "tubit23" in Tutorium "t01" und Gruppe "g01". Analog dazu werden Sie in Ihrem SVN Repository eine Struktur finden, in der Sie – wie auch schon im C-Kurs – Verzeichnisse zum Abgeben Ihrer Lösung (`submission/`), für Feedback von uns (`feedback/`) und zum Testen Ihres Codes (`workdir/`) finden. Diese Struktur befindet sich sowohl in Ihrem persönlichen als auch in Ihrem Gruppenordner.

1. Aufgabe: Kleinstes Element einer Liste finden (20 Punkte)

1.1. (10 Punkte) Beschreiben Sie mittels Pseudocode einen Algorithmus, der das kleinste Element einer Liste (von Zahlen) findet und ausgibt.

1.2. (10 Punkte) Implementieren Sie anhand des von Ihnen entwickelten Pseudocodes aus der vorherigen Aufgabe die Funktion `finde_minimum()` in C. Die Funktion bekommt als Eingabe ein Integerarray und gibt die kleinste Zahl zurück.

Die Liste von Integer Zahlen soll über eine Dateiumleitung (Redirection) aus einer Datei mitgegeben werden. Das ist genau so als würden die Zahlen über die Standardeingabe einzeln eingegeben werden und die Eingabe mit Ctrl-D beendet.

```
./introprog_blatt01_aufgabe01_2 < zahlen.txt
```

Benutzen Sie die unten stehende Code-Vorgabe.

Listing 1: Vorgabe semester_blatt01_aufgabe01_2_vorgabe.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int MAX_LAENGE = 1000;
5
6 // Gibt das Array nach dem gewünschten Format aus.
7 void print_array(int array[], int len) {
8     printf("Liste:");
9     for (int i = 0; i < len ; i++) {
10         printf("_%d", array[i]);
11     }
12     printf("\n");
13 }
14
15 int lese_array(int array[]) {
16     int i = 0;
17
18     while (scanf("%d", &array[i]) == 1) {
19         i++;
20     }
21     return i;
22 }
23
24 // Finde den kleinsten Wert des Array und gebe ihn zurück
25 int finde_minimum(int array[], int len) {
26     // HIER
27     // Finde das Minimum der Liste und gebe es zurück. Halte dich soweit
28     // ↳ möglich an
29     // deinen Pseudo Code.
30     // Rückgabewert ist das Minimum
31 }
32
33 int main(int argc, char *argv[]) {
34     int array[MAX_LAENGE];
35     int len = lese_array(array);
36     print_array(array, len);
37     //HIER rufe finde_minimum auf
38     int minimum = ...
39     printf("Kleinsten_Wert:_%d\n", minimum);
40     return 0;
41 }
```

2. Aufgabe: Implementierung Insertion Sort (20 Punkte)

Implementieren Sie anhand des Pseudocodes, der in der Vorlesung vorgestellt wurde die Funktion `insertion_sort()` in C. Die Funktion bekommt als Eingabeparameter ein Integer-Array, das sortiert zurückgegeben werden muss. Das Programm wird wie folgt aufgerufen:

```
./introprog_blatt01_aufgabe02 < zahlen.txt
```

Wie auch in der letzten Aufgabe soll Ihr Programm die Integer Zahlen via einer Dateiumleitung (Redirection) entgegennehmen. Diesen Teil des Programms finden Sie wieder in der Vorgabe. Sie finden eine Beispielliste im Verzeichnis `zahlen.txt` in ISIS. Die Werte in unserer Beispieldatei dienen nur als Beispiele. Geben Sie sowohl die Eingabe- als auch die Ergebnisliste mittels der vorgegebenen Funktion `print_array` aus.

Benutzen Sie folgende Codevorgabe:

Listing 2: Vorgabe semester_blatt01_aufgabe02_vorgabe.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int MAX_LAENGE = 1000;
5
6 // Gibt das Array nach dem gewünschten Format aus.
7 void print_array(int array[], int len) {
8     printf("Liste:");
9     for (int i = 0; i < len ; i++) {
10         printf("_%d", array[i]);
11     }
12     printf("\n");
13 }
14
15 int lese_array(int array[]) {
16     int i = 0;
17
18     while (scanf("%d", &array[i]) == 1) {
19         i++;
20     }
21     return i;
22 }
23
24 void insertion_sort(int array[], int len) {
25     // HIER
26     // Wir sortieren das Array nach dem Insertion sort Verfahren. Dabei
27     // ↳ benutzen wir die
28     // in-place Variante, d.h. wir benutzen nur das originale Array.
29 }
30
31 int main(int argc, char *argv[]) {
32     int array[MAX_LAENGE];
33     int len = lese_array(array);
34     printf("Unsortierte_"); print_array(array, len);
35     // HIER
36     // Rufe die insertionsort Funktion auf.
37     printf("Sortierte_"); print_array(array, len);
38     // Nicht vergessen den Speicher auch wieder freizugeben :).
39     return 0;
40 }
```