

Universität Potsdam
Institut für Computational Science und Informatik
Sommersemester 2015

Praxis der Programmierung

Aufgabe für Woche 10

Diese Aufgabe Stützt sich auf die 3. Aufgabe von Woche 8, dem Melderegister im Einwohner Meldeamt. Schreiben Sie nun ein C-Programm *UseRegister*, dass das Melderegister lediglich Benutzt.

1. Kopieren Sie die *Einwohnermeldeamt.c* in Ihr Arbeitsverzeichnis. Entfernen Sie alle Deklarationen vor den Definitionen (bzw Schreiben Sie diese getrennt) und fassen Sie alle Deklarationen in einer Header-Datei zusammen. Binden Sie diese mit einer Präprozessor-Anweisung ein (Wie in Woche 4), so dass das Programm ablauffähig ist und das gleiche Verhalten wie vorher zeigt.
2. Entfernen Sie Die Automatische Befüllung des Melderegisters, sowie das Anlegen eines Beispiel Einwohners *Sherlock Holmes*.
3. Schreiben Sie jetzt ein C-Programm *UseRegister*, dass folgende Funktion erfüllt.
 - (a) Das Programm fragt in einer Endlosschleife Eingaben vom Nutzer ab. Diese Schleife wird mit der Eingabe 0 beendet. Es wird zudem eine Datenbank erstellt (ein Array mit Einwohnern). Dabei soll die Größe der Datenbank vom Nutzer eingegeben werden.
 - (b) Mit der Eingabe 1, wird eine Funktion Aufgerufen, die die Datenbank automatisch befüllen soll.
 - (c) Mit der Eingabe 2, wird eine Funktion Aufgerufen, die eine Manuelle Eingabe eines Neuen Einwohners in die Datenbank ermöglicht. Dabei soll darauf geachtet werden, dass es keine zwei Einwohner mit gleichem (Vor- und) Nachnamen geben darf.
 - (d) Mit der Eingabe 3, wird eine Funktion Aufgerufen, mit der die Datenbank nach einem Namen durchsucht werden kann. Falls es keinen Einwohner dieses Namens gibt, wird das Ergebnis auf der Konsole ausgegeben, ansonsten wird der Datenbankeintrag mit der in der letzten Woche erstellten Ausgabefunktion ausgegeben.
 - (e) Mit der Eingabe 4, wird eine Funktion Aufgerufen, mit der die Datenbank nach Namen Sortiert werden soll.
 - (f) Mit der Eingabe 5, wird eine Funktion Aufgerufen, mit der die Datenbank wieder gelöscht wird.

Mergesort

4. Implementieren Sie den hier beschriebenen Algorithmus.

Kopieren Sie dazu die Datei *mergesort.c* und bearbeiten Sie diese.

Mergesort ist ein Divide-and-Conquer-Sortierverfahren. Man betrachtet die zu sortierenden Daten als Liste und zerlegt sie in kleinere Listen, die jede für sich sortiert werden. Die sortierten kleinen Listen werden dann im Reißverschlussverfahren zu größeren Listen zusammengefügt (engl. to merge), bis wieder eine sortierte Gesamtliste erreicht ist.

Eine Liste der Länge n wird solange rekursiv geteilt, bis n einelementige Teillisten entstehen. Nun folgt der eigentliche Sortiervorgang: Man nimmt immer zwei Teillisten und fügt diese zu einer neuen sortierten Teilliste zusammen, indem man immer das kleinste Element der beiden Listen streicht und in die neue Teilliste einfügt. Dieses Verfahren muss dann solange wiederholt werden, bis es nur noch eine Teilliste gibt. Die folgende Abbildung zeigt an einem Beispiel, wie der Algorithmus arbeitet:

