Übungsblatt 4

Willkommen zum Praktikum zu Programmieren 3.

Aufgabe 1. Schreiben Sie ein Programm `abzaehl`, das die Auswahl einer Person aus einer Gruppe von Personen mit Hilfe eines Abzählreims simuliert. Als Kommandozeilenparameter erhalten Sie die Anzahl der Silben des Abzählreims (z.B. 7 für "Ene mem mu und raus bist du") und dann alle Namen der Personen in der Reihenfolge Ihres Auftretens. Man beginnt das Auszählen mit der ersten Person.

Lesen Sie die Namen in einem dynamisch erzeugten Ring (eine geschlossene Liste) ein und simulieren Sie das Abzählen durch weiterrücken im Ring und das Entfernen aus der Gruppe durch das Löschen des aktuellen Elements im Ring. Stellen Sie sicher, dass Sie keine Speicherlecks haben.

Aufgabe 2. Schreiben Sie eine Bibliothek für doppelt verkettete Listen. Der Datensatz kann beliebig sein (Zeiger auf **void**). Die Speicherverwaltung für die Daten obliegt dem Rufer. Repräsentieren Sie die Liste durch einen Zeiger auf das erste Element oder `NULL` wenn die Liste leer ist. Realisieren Sie die Funktionen

```
lptr add_first(lptr l, void *data);
lptr add_ith(lptr l, int i, void *data);
lptr add_last(lptr l, void *data);
```

wobei jeweils das erste Argument die Liste, das letzte Element die Daten sind und respektive an erster, i.ter, bzw. letzter Stelle eingefügt wird. Es wird jeweils die geänderte Liste zurückgegeben. Schreiben Sie Funktionen zum Löschen von Elementen

```
lptr del_first(lptr l);
lptr del_ith(lptr l, int i);
lptr del_last(lptr l);
```

mit je erstem Parameter die Liste und Rückgabewert die geänderte Liste. Realisieren Sie auch die Funktionen

```
void del_all(lptr l)
void del_all_custom(lptr l, void (*custom)(void *))
```

`del_all` löscht die übergebene Liste, während `del_all_custom` zusätzlich für jeden Datensatz vor dem Löschen die Funktion `custom` mit den Daten des Listenelements aufruft. Realisieren Sie auch die Funktionen

```
lptr copy(lptr l)
lptr copy_custom(lptr l, void *(*custom)(void *)) ,
```

die Listen kopieren. Bei `copy` wird die Adresse des Datensatzes einfach kopiert. Bei `copy_custom` wird zusätzlich die Adresse des Datensatzes auf den Rückgabewert des Aufrufs von `custom` mit dem aktuellen Datensatz gesetzt.

Testen Sie Ihre Bibliothek mit einem Programm `listargs`, das eine List aus allen Kommandozeilenparametern aufbaut. Kopieren Sie die Liste einmal geeignet mit `copy` und einmal mit `copy_custom`. Geben Sie die Strings in der Liste aus.

Aufgabe 3. Schreiben Sie ein Programm `mergesort`, das die Kommandozeilenparameter alphabetisch mit dem MergeSort-Verfahren sortiert. Verwenden Sie die Listen-Bibliothek aus Aufgabe 2. Implementieren Sie den MergeSort so, dass eine neue Liste zurückgegeben wird in der sowohl die Listenstruktur als auch die Zeichenketten kopiert wurde. Geben Sie die sortierte Liste aus.

Realisieren Sie den MergeSort-Algorithmus so, dass er für beliebige Daten verwendet werden kann. Lagern Sie die Implementierung des MergeSorts in eine separate Übersetzungseinheit aus.

Aufgabe 4. Schreiben Sie ein Programm `mergesortf`, das wie das Programm aus Aufgabe 3 Listen von Zeichenketten sortiert. Allerdings ist die Eingabe diesmal nicht die Kommandozeilenparameter sondern eine Datei. Eine Datei wird angegeben mit der Option `-f <dateiname>`. Wenn keine Datei angegeben wird, dann wird die Standard-Eingabe verwendet. Es wird zeilenweise eingelesen. Die sortierte Datei wird wieder auf der Standardausgabe ausgegeben.

Stellen Sie sicher, dass Sie mit beliebig langen Zeilen in der Eingabedatei umgehen können. Schreiben Sie sich dazu eine Funktion `read_line`, die eine Zeile einliest und einen Zeiger auf einen mindestens passend großen im Freispeicher allokierten Speicherbereich mit dem Inhalt der nächsten Zeile zurück gibt, bzw. `NULL` wenn das Ende der Datei erreicht wurde.

Hinweis 1. Verwenden Sie in den Aufgaben 2, 3 und 4 Makefiles mit separater Übersetzung. Erweitern Sie die Listen-Bibliothek nach Bedarf. Stellen Sie sicher, dass Sie keine Speicherfehler haben. Versuchen Sie effiziente Programme zu schreiben.

Hinweis 2. Verwenden Sie die C-Standardbibliothek sinnvoll. Versuchen Sie selbst passenden Funktionen, die Sie bisher noch nicht kennen, zu finden und verwenden Sie sie.

`http://www.mi.hs-rm.de/~barth/hsrm/prog3`