



**Aufgabe 1.** Implementiere doppelt verkettete Listen, die beliebige Daten speichern können (als `void *`). Die Liste soll zusätzlich einen Stack enthalten, auf dem gelöschte Elemente abgelegt werden, damit man sie wiederherstellen kann.

```
1  /* Leere Liste erstellen */
2  LIST *list_create();
3
4  /* Element hinter E einfügen, NULL heißt am Anfang */
5  LISTNODE *list_insert(LIST *L, LISTNODE *E, void *p);
6  /* Element hinter E einfügen, dessen datenpointer
7     auf n Zellen allokierten Speicher zeigt. Sollte bei
8     Speichermangel einen Nullpointer liefern. */
9  LISTNODE *list_insert_alloc(LIST *L, LISTNODE *E,
10     unsigned int n);
11
12 /* Element am Anfang bzw. Ende entfernen und
13    die Daten zurück geben */
14 void *list_shift(LIST *L);
15 void *list_pop(LIST *L);
16
17 /* eine Element aus der Liste entfernen */
18 void list_delete(LIST *L, LISTNODE *E);
19
20 /* Liste inklusive allen Elementen frei geben */
21 void list_free(LIST *L);
22
23 /* Das eben gelöschte Element wiederherstellen */
24 void list_undelete(LIST *L);
25 /* Alle gelöschten Elemente wiederherstellen */
26 void list_dance(LIST *L);
```

**Aufgabe 2.** Schreibe ein Programm, das Worte aus einer Datei einliest, diese in eine Liste einfügt, anschließend die Liste lexikographisch sortiert und dann die Worte sortiert ausgibt.