Aufgabe 1. Schau dir die Folien zu Vorlesung 7 nochmal an und implementiere doppelt verkettete Listen, die `double`-Variablen speichern.

Schreibe einen Testcode, mit dem du überprüfen kannst, dass deine Implementation doppelt verketteter Listen richtig arbeitet.

```
/* Definiere hier angemessene Strukturen fuer einen
   einzelnen Listeneintrag und die Liste selbst. */

/* Leere Liste erstellen */
LIST *list_create();

/* Element hinter E einfuegen, NULL heisst am Anfang */
LISTNODE *list_insert(LIST *L, LISTNODE *E, double p);

/* Element am Anfang bzw. Ende einfuegen */
LISTNODE *list_unshift(LIST *L, double p);
LISTNODE *list_push(LIST *L, double p);

/* Element am Anfang bzw. Ende entfernen und
   die Daten zurueck geben */
double list_shift(LIST *L);
double list_pop(LIST *L);

/* ein Element aus der Liste entfernen */
void list_delete(LIST *L, LISTNODE *E);

/* zwei Listen zusammenfuegen */
LIST *list_merge(LIST *L, LIST *M);

/* Liste inklusive allen Elementen frei geben */
void list_free(LIST *L);
```

Aufgabe 2. Implementiere eine Funktion die zu einem gegebenen Funktionenpointer $f : \mathbb{R} \to \mathbb{R}$, einen Dateinamen, einer Schrittweite $s \in \mathbb{R}$, einer Startstelle x_1 und einer Endstelle x_2 die Wertetabelle der Funktion zwischen x_1 und x_2 zur Schrittweite s speichert. Dabei sollen x und $f(x)$ durch einen Tabulator getrennt werden und jedes Paar $(x, f(x))$ in einer eigenen Zeile stehen.

Aufgabe 3. In dieser Aufgabe geht es um numerische Integration.

a) Implementiere eine Integrationsfunktion, die das Intervall $[a, b]$ in n gleich große Teile aufteilt, für diese jeweils die Trapezsumme (aus dem Skript) berechnet und diese aufsummiert:

```
double integrate(double a, double b,
  double (*f)(double), unsigned int n);
```

b) Schreibe nun eine Funktion, die nicht die Anzahl der Teilintervalle erhält, sondern eine “Fehlertoleranz” e. Die Funktion die Aufteilung solange verfeinern, bis sich der approximierte Wert für das Integral durch eine Verfeinerung nur noch um weniger als e ändern würde.