Aufgabe 1. Lese noch einmal im Skript die Sektion 7.5 und implementiere doppelt verkettete Listen, die `double`-Variablen speichern.

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

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

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

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

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

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

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

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

Aufgabe 2. 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 Intervalle $[a_i, b_i]$ jeweils die Trapezsumme $\frac{(f(b_i)+f(a_i))(b_i-a_i)}{2}$ 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.

Aufgabe 3.

a) Implementiere die Addition, Multiplikation, Potenzen und Division komplexer Zahlen. Verwende dazu folgende Header-Datei:

```
#ifndef _COMPLEX__H
#define _COMPLEX__H

typedef struct _COMPLEX {
        double real;
        double imag;
} COMPLEX;

COMPLEX cplx_add(COMPLEX a, COMPLEX b);
COMPLEX cplx_mul(COMPLEX a, COMPLEX b);
COMPLEX cplx_pow(COMPLEX a, unsigned long n);
COMPLEX cplx_div(COMPLEX a, COMPLEX b);

#endif
```

b) Implementiere die Addition, Multiplikation und Division sowie das Kürzen rationaler Zahlen. Schreibe dazu erst die Header-Datei.