

Aufgabe 1. Nimm dir bitte 10 Minuten Zeit und fülle unsere Umfrage zu der Vorlesung aus. Selbstverständlich ist das anonym. Den Link zu der Umfrage findest du auf unserer Webseite:

Aufgabe 2. Erledige nun erstmal die Aufgaben der anderen Zettel.

Aufgabe 3. Implementiere eine Funktion die zu einem gegebenen Funktionspointer $f : \mathbb{R} \rightarrow \mathbb{R}$, einem 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. Etwa wäre die Ausgabe für $f = \cos$ zwischen $x_1 = 0$ und $x_2 = 0$ mit Schrittweite $s = 0.1$ die folgende:

```
1 0.0    1.0
2 0.1    0.995004165278
3 0.2    0.980066577841
4 0.3    0.955336489126
5 0.4    0.921060994003
6 0.5    0.87758256189
7 0.6    0.82533561491
8 0.7    0.764842187284
9 0.8    0.696706709347
10 0.9    0.621609968271
11 1.0    0.540302305868
```

Aufgabe 4. Lese noch einmal im Skript die Sektion 7.5 und implementiere doppelt verkettete Listen, die `double`-Variablen speichern.

```
1 /* Definiere hier angemessene Strukturen fuer einen
2    einzelnen Listeneintrag und die Liste selbst. */
3
4 /* Leere Liste erstellen */
5 LIST *list_create();
6
7 /* Element hinter E einfuegen, NULL heisst am Anfang */
8 LISTNODE *list_insert(LIST *L, LISTNODE *E, double p);
9
10 /* Element am Anfang bzw. Ende einfuegen */
```

```
11 LISTNODE *list_unshift(LIST *L, double p);
12 LISTNODE *list_push(LIST *L, double p);
13
14 /* Element am Anfang bzw. Ende entfernen und
15   die Daten zurueck geben */
16 double list_shift(LIST *L);
17 double list_pop(LIST *L);
18
19 /* ein Element aus der Liste entfernen */
20 void list_delete(LIST *L, LISTNODE *E);
21
22 /* zwei Listen zusammenfuegen */
23 LIST *list_merge(LIST *L, LIST *M);
24
25 /* Liste inklusive allen Elementen frei geben */
26 void list_free(LIST *L);
```