Hausübung 2 (Teil B der Abgabe), 30min Bearbeitung, MI1545

1. Laden Sie sich die Quelldatei `task.c` aus TUWEL herunter.
2. Passen Sie die Quelldatei nach den Anforderungen der Aufgabenstellung an.
3. Abgabe: Laden Sie die abgeänderte Quelldatei wieder als `task.c` in TUWEL hoch.

- Sie dürfen keine andere Quelldatei öffnen.
- Sie dürfen keinen Quellcode aus anderen Quellen/Dateien kopieren.
- Sie dürfen *googeln*; aber auch im Browser dürfen Sie keine anderen Quelldateien öffnen oder betrachten.

Gegeben ist die Implementierung `struct Vector` (*Vector*) analog zur Hausübung (`Vector.h/Vector.c`). Eine Beschreibung der Struktur samt zugehöriger Funktionen steht in `Vector.pdf` zur Verfügung.

Erweitern Sie das Programm in `task.c` mit der Implementierung der unten spezifizierten Funktion und erweitern Sie die `main`-Funktion wie unten beschrieben.

Empfohlener Kompilierbefehl/Ausführen:

```
gcc -std=c11 -fsanitize=address -Wall -pedantic -g Vector.c task.c -lm && ./a.out
```

Funktion `num_interpolate`: Funktionswerte interpolieren (8 Punkte)

```
struct Vector *num_interpolate(const struct Vector *xinterp, const struct Vector *x,
                               const struct Vector *f);
```

Funktionsbeschreibung: Interpoliert 1D-Funktionswerte an den übergebenen Stellen (`xinterp`) linear mittels ebenfalls übergebener diskreter Funktionswertepaare (`x`, `f`) und speichert die interpolierten Funktionswerte in ein dynamisch alloziertes Vector-Object. Zu interpolierende Werte ausserhalb des durch `x` vorgegebenen Intervals nehmen den ensprechenden Randwert an.

Rückgabewert: Zeiger auf das dynamisch allozierte Vector-Objekt.

Parameter:

- `xinterp`: Zeiger auf den Vector mit x-Werten für die der Funktionswert interpoliert wird.
- `x`: Zeiger auf den Vector mit x-Werten.
- `f`: Zeiger auf den Vector mit zugehörigen Funktionswerten.

Annahmen:

- Die x-Koordinaten im übergebenen Vector `x` sind monoton aufsteigend.
- Die Vectoren `x` und `f` enthalten dieselbe Anzahl an Elementen.

Anwendung in `main`: Funktion abtasten und Funktionswerte interpolieren (7 Punkte)

Gegeben sind die Funktionen `num_linspace`, `num_sample` und die Function `func`. In der `main`-Funktion werden mittels der Funktion `num_linspace` 10 äquidistante Werten im Intervall `[-2.0,0.0]` erzeugt (Variable `x`). Ebenso werden 5 äquidistante Werten im Intervall `[-3.0,1.0]` erzeugt (Variable `xinterp`).

Erweitern Sie die Funktionalität in der `main`-Funktion um Folgendes:

1. Tasten Sie die Funktion `func` an den 10 gegebenen x-Werten (Variable `x`) mittels der Funktion `num_sample` ab.
2. Interpolieren Sie mittels der Funktion `num_interpolate` die Funktionswerte an den 5 gegebenen x-Werten (Variable `xinterp`).
3. Geben Sie die 10 abgetasteten Funktionswerte und ebenso die 5 interpolierten Funktionswerte mittels der Funktion `vector_print` in der Konsole aus.