

Übungen zur Computerorientierten Physik

4 Listen (Suchen, Löschen)

1. Laden Sie das Programm `list_fragment.c` vom StudIp.
2. Implementieren Sie eine Funktion zum Suchen von Listenelementen

```
/****** search_info() *****/
/** Search for 'value' in 'list' */
/** PARAMETERS: (*)= return-paramter */
/**          list: first element of list */
/**          value: to be found */
/** RETURNS: */
/**  pointer to element (NULL if not found) */
/******
elem_t *search_info(elem_t *list, int value)
```

(4 Punkte)

3. Implementieren Sie eine Funktion zum Entfernen von Listenelementen

```
/****** remove_element() *****/
/** Remove 'elem' from 'list' */
/** PARAMETERS: (*)= return-paramter */
/**          list: first element of list */
/**          elem: to be removed */
/** RETURNS: */
/**  (new) pointer to beginning of the list */
/******
elem_t *remove_element(elem_t *list, elem_t *elem)
```

Hinweis: In der Funktion müssen Sie sich zuert das Element vor `elem` besorgen.

Hinweis 2: Das Element `elem` darf nach dem Entfernen keinen Nachfolger haben.

Hinweis 3: Die Funktion muss auch funktionieren, wenn das `elem` nicht in der Liste enthalten ist!

(6 Punkte)

4. Führen Sie die Funktionen mit geeignet konstruierten Beispielen vor, insbesondere, dass alle möglichen Fälle (z.B. Entfernen am Anfang, Entfernen eines Elementes, das nicht in der Liste ist, etc. analog für Suchen) abgedeckt sind, genauer:

- ☐ Suchen eines Elementes, das vorkommt
- ☐ Suchen eines Elementes, das nicht vorkommt
- ☐ Suchen in einer leeren Liste
- ☐ Entfernen eines Elementes am Anfang der Liste
- ☐ Entfernen eines Elementes in der Mitte der Liste
- ☐ Entfernen eines Elementes am Ende der Liste
- ☐ versuchtes Entfernen eines Elementes, das nicht vorkommt

Lassen Sie sich die Liste jeweils vor und nach Ausführen der Operation ausgeben.