1 Bibliothek zum Arbeiten mit einfach-verketteten Listen

Aufgabe war es eine C-Bibliothek zu entwickeln mit der man einfach verkettete Listen erstellen, bearbeiten und löschen kann.

**Node\_t\* createNode(value\_t value**

Hier wird einfach mit dem übergebenen Wert ein neuer node erzeugt und zurückgegeben.

**Node\_t\* getNode(node\_t\* pHead, int index)**

Hier wird von einer gegeben Liste eine Node zurück geliefert. Der index gibt an an welcher stelle wir den Node gefunden haben.

**int insertFront(node\_t\*\* ppHead, const value\_t\* pValue)**

HIer wird am Anfang der Liste, mithlfe von pointer umbiegen, ein neu erstellter Node mit einem bestimmten Wert eingefügt.

**int insertBefore(node\_t\*\* ppHead, node\_t\* pPrevNode, const value\_t\* pValue)**

Hier wird ein Node vor dem **PrevNode** eingefügt. Zuerst wird hier die position des prevNode gesucht. Dannach wird ein neuer Node vor dem **PrevNode** eingefügt.

**int deleteValues(node\_t\*\* ppHead, const value\_t\* pValue, int deleteAllOcc)**

Wenn deleteAllOcc false ist wird der erste Node gesucht bei dem der Wert übereinstimmt. Dieser wird dann raus gelöscht (Speicher wird dabei freigegeben).   
Wenn deleteAllOcc True ist werden alle Nodes mit diesem Wert gelöscht (Speicher wird dabei freigegeben).

**int deleteList(node\_t\*\* ppHead)**

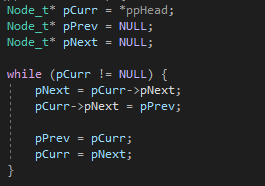
Hier wird solange über die Liste iteriert (while (\*ppHead != NULL)) bis alle gelöscht sind.

**void printList(const node\_t\* pHead)**

Hier wird über die Liste drüber iteriert (while (pCurr != NULL)) und bei jeder Iteration wird der aktuelle Node ausgegeben

**int inverseList(node\_t\*\* ppHead)**

Hier wird die ganze Liste invertiert indem man einfach in einer while Schleife solange die Pointer umbiegt bis pCur == NULL ist.



**int sortList(node\_t\*\* ppHead)**Hier werden alle Nodes mit dem Selection sort Algorithmus sortiert.

**Testfälle**:  
siehe main.c file.

C-Program code: Siehe beigelegtes linkedList.c/.h file.