

Übungsblatt 2

Implementieren Sie eine doppelt-verkettete Liste zum sortierten Speichern von Integerwerten. Die Liste soll aus folgenden Strukturen aufgebaut werden:

```
/* Datenstruktur eines Listenelements */
typedef struct eintrag {
    int key;
    struct eintrag * p_prev;
    struct eintrag * p_next;
} eintrag_t;

/* Datenstruktur mit Informationen ueber die Liste */
typedef struct { eintrag_t * p_head;
                 eintrag_t * p_tail;
                 eintrag_t * p_curr;
                 int length;
} liste_t;
```

Der Einfachheit halber sollen die key-Werte eindeutig sein. Implementieren Sie mindestens folgende Funktionen:

```
//Legt den Speicher für ein Listenelement an,
//speichert den key-Wert im Element.
//Gibt einen Zeiger auf das neues Listenelement zurück.
eintrag_t * create_element (int key);

//Fügt ein Element sortiert in die Liste ein.
//Zuerst wird die Position gesucht, an der das Listenelement
//eingefügt werden soll.
//Es sind 3 Möglichkeiten zu unterscheiden, am Anfang der Liste,
//mitten in der Liste oder am Ende der Liste.
//Anschließend wird das Element in die Liste eingefügt.
void insert_element(liste_t * liste, eintrag_t * p_insert);

//Gibt alle Listen-Elemente aus
void print_list(liste_t liste);
```

Falls Sie nicht sicher im Umgang mit Zeigern sind, implementieren Sie zuerst eine einfach-verkettete Liste

```
/* Datenstruktur eines Listenelements */
typedef struct eintrag {
    int key;
    struct eintrag * p_next;
} eintrag_t;
```

Für die einfach-verkettete Liste finden Sie Hinweise zum Programm auf den Folien von PE1 im Kapitel *Komplexe Datenstrukturen* und auf meinen ALD-Folien.

Falls Sie sich sicher im Umgang mit Zeigern fühlen, erweitern Sie die Liste, so dass nicht nur ein Integer in jedem Element abgespeichert wird, sondern eine Datenstruktur für einen Telefonbucheintrag:

```
/* Datenstruktur der Daten eines Listenelements */
typedef struct {
    char * n_name;
    char * v_name;
    int  nummer;
} person_t;
```

Die Liste soll dann nach den Namen sortiert werden (in C Vergleiche mit Hilfe der Funktion `strcmp`). Falls Sie bereits Kenntnisse in C++ haben, verwenden Sie Klassen anstatt Strukturen und die Klasse `string` anstatt des Datentyps `char *`.