

Übungsblätter zur Vorlesung Algorithmen und Datenstrukturen  
(Magister)  
Sommersemester 2002

(Prof. Der)  
Serie 3

Abgabetermin: 31.05. 2002 vor der Vorlesung

---

1. Aufgabe Datenmodell Liste (2 Pkte): Eine nichtleere Liste besteht aus ihrem ersten Element und dem Rest der Liste. Wenn der Rest des Restes einer Liste  $L$  leer ist, welche Länge hat dann die Liste  $L$ ?
2. Aufgabe (je Teilaufgabe 1 Pkt): Gegeben sei die Liste (1, 6, 0, 7, 1).
  - (a) Welche Länge und wieviele Positionen hat die Liste?
  - (b) Wie lauten alle Teillisten?
  - (c) Wieviele Teilsequenzen gibt es (mit Begründung)?

3. Gegeben seien folgende Definitionen

```
typedef struct LE{                                struct LE *next;
                                                    struct LE *previous;
                                                    char  *Inhalt
}ListElmt;
typedef struct {                                  int  size;
                                                    ListElmt *head;
                                                    ListElmt *tail;
}Liste;
Liste *MeineListe;
ListElmt *element, *new_element;
```

MeineListe->head ist beispielsweise nun der Zeiger auf den Kopf der Liste mit Namen MeineListe, analog ist element->next der Zeiger auf das auf \*element (das Element mit Zeiger element) folgende Element. MeineListe->head und MeineListe->tail seien hier dummy Elemente. Aufgaben (Je 2 Pkte. plus ein Zusatzpunkt für syntaktische Korrektheit, d.h. mit Speicheranforderung):

- (a) Initialisierung der Liste (Zeiger auf NULL und size auf 0 setzen)
- (b) Angenommen die Liste habe schon die Länge  $n$ . Fügen Sie am Ende der Liste ein neues Element (mit Zeiger new\_element) ein.
- (c) Fügen Sie das neue Element vor dem ersten (nicht-dummy) Element ein.
- (d) Fügen Sie das neue Element hinter \*element (d.h. nach dem Element mit Zeiger element) ein.

- (e) Die Liste sei doppelt verkettet. Fügen Sie nach \*element ein neues Element ein.
4. Aufgabe: (6 Punkte) Ihnen steht ein Stapel (Stack) mit den Standardoperationen top, push und pop zur Verfügung. Geben Sie die Folge der Operationen und die jeweilige Belegung des Stacks an, die sich bei der Abarbeitung folgender Postfixausdrücke ergeben: (Die Operatoren \* und + bzw. - haben jeweils genau zwei Operanden.)
- (a)  $a b + c d * + e *$
- (b)  $a b c d + + +$
- (c)  $a b + c + d + e +$
5. Aufgabe (2 Pkte): Wie sieht ein Stack nach Ausführung der folgenden Operationen aus? push(a), push(b), pop, push(c), push(d), pop, push(e), pop, pop. Start mit dem leeren Stack.
6. Aufgabe (2 Pkte): Überführen Sie den Ausdruck

$$3 * (7 + 4) / (2 + 11) - 3$$

in die Postfixschreibweise. Dabei ist auf die unterschiedliche Bindungsstärke der Operatoren zu achten.