Prof. Dr. A. Weber
Hassan Errami

Universität Bonn
Institut für Informatik II
06.12.2013

Wintersemester 2013

Übungen zur Vorlesung
Algorithmisches Denken und imperative Programmierung (BA-INF-014)
Aufgabenblatt 6

Zu bearbeiten bis: 13.12.2013

Aufgabe 1 *(Listen 1 - 8 Punkte)*

a) Die Funktionen *insertFirst*, *reverseListCon* und *reverseList* seien wie in den Vorlesungsfolien implementiert. Gegeben sei folgende main-Funktion:

```
int main(int argc, char *argv[])
{
IntNode *L2, *L1 = NULL;
L1= insertFirst( L1, 5);
L1= insertFirst( L1, 6);
L2= reverseListCon(L1);
// Stelle1

reverseList(L1);
//Stelle 2

L1= insertFirst( L1, 3);
L2= insertFirst( L2, 4);
//Stelle 3

return 0;
}
```

Skizzieren Sie den Zustand des Speichers an den Stellen 1, 2 und 3!

b) In der Aufgabe 2 des letzten Übungsblattes haben Sie die Datenstruktur *DoubleNode* über *Double* Zahlen implementiert.

- Implementieren Sie die Funktion *DoubleNode * insertLast(DoubleNode *head, double d)* zum Einfügen von *double d* am Ende einer Liste.
- Implementieren Sie die Funktion *DoubleNode * reverseDoubleListCon(DoubleNode *head)* zum konstruktiven Invertieren einer Liste.
- Implementieren Sie die Funktion *DoubleNode * reverseDoubleList(DoubleNode *head)* zum destruktiven Invertieren einer Liste.

Aufgabe 2 *(Listen 2 - 8 Punkte)*

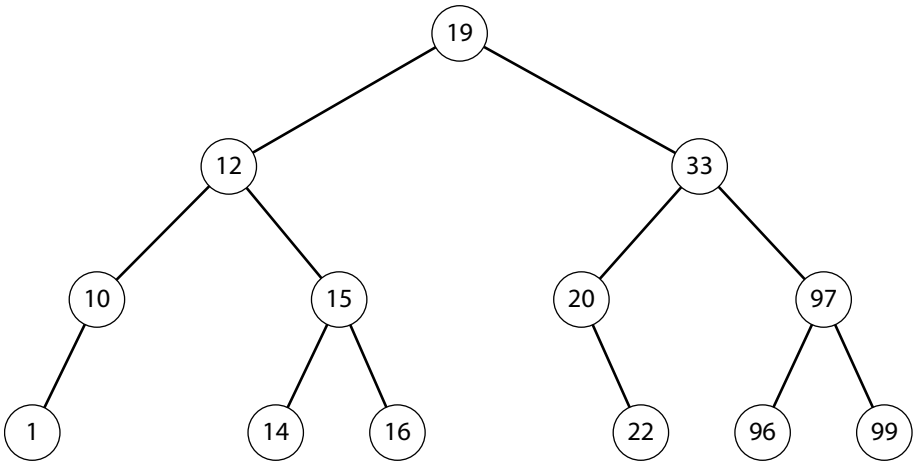
Implementieren Sie folgende Operationen auf Listen.

- get(int i), die das i-te Element der Liste zurückgibt,
- delete(int i), die das i-te Element aus der Liste entfernt,
- insert(T *elem*, int i), die das Element *elem* an der i-ten Stelle der Liste einfügt.

Analog zur Array-Indizierung sei hierbei i = 0 die Position des ersten Listenelements. Behandeln Sie jeweils die Ausnahme, dass der Index i die Länge der Liste übersteigt!

Aufgabe 3 *(Baumdurchläufe - 4 Punkte)*

Gegeben sei folgender Binärbaum:



a) Geben Sie die Elemente des Baumes in der Reihenfolge an, in der sie bei einem *preorder*-Durchlauf bearbeitet werden!

b) Geben Sie die Elemente des Baumes in der Reihenfolge an, in der sie bei einem *postorder*-Durchlauf bearbeitet werden!

c) Geben Sie die Elemente des Baumes in der Reihenfolge an, in der sie bei einem *inorder*-Durchlauf bearbeitet werden!