

Wintersemester 2013  
**Übungen zur Vorlesung**  
**Algorithmisches Denken und imperative Programmierung (BA-INF-014)**  
**Aufgabenblatt 5**  
Zu bearbeiten bis: 06.12.2013

**Aufgabe 1** (*Listen - 10 Punkte*)

Sie haben in der Vorlesung die Datenstruktur *IntNode* für Listen über *Integer* Zahlen sowie Funktionen für die Listenoperationen kennengelernt.

a) Sei folgende main-Funktion gegeben:

```
int main(int argc, char *argv[])
{
    IntNode *L2, *L1 = NULL;
    L1= insertFirst( L1, 3);
    L1= insertFirst( L1, 7);
    L1= insertFirst( L1, 11);
    L2=L1;
    L2= insertFirst( L2, 5);
    L2= insertFirst( L2, 14);

    \\ Stelle1

    printList(L1);
    printList(L2);
    return 0;
}
```

Skizzieren Sie den Zustand des Speichers an **Stelle 1**.

b)

- Implementieren Sie entsprechende Datenstruktur *DoubleNode* für Listen über *Double* Zahlen.
- Implementieren Sie die Funktion *DoubleNode \* insertFirst( DoubleNode \*head, double c)* zum Einfügen von *double d* am Anfang einer Liste.
- Implementieren Sie die Prozedur *void printList(DoubleNode \*head)*, die den Inhalt einer Liste auf der Konsole ausgibt.

**Aufgabe 2** (*Structs - 10 Punkte*)

Komplexe Zahlen können in der Form  $a + b \cdot i$  dargestellt werden, wobei a und b reelle Zahlen sind und i die imaginäre Einheit ist. Auf die so dargestellten komplexen Zahlen lassen sich die üblichen Rechenregeln für reelle Zahlen anwenden, wobei  $i^2$  stets durch -1 ersetzt werden kann und umgekehrt.

Mehr zu komplexen Zahlen finden Sie unter [http://de.wikipedia.org/wiki/Komplexe\\_Zahl#](http://de.wikipedia.org/wiki/Komplexe_Zahl#).

a) Schreiben Sie einen Datentyp (*Complex*) zur Speicherung von komplexen Zahlen. Dieser soll Real- und Imaginärteil jeweils als double speichern.

b) Implementieren Sie die Funktionen *add*, *sub*, *mult* und *div* zur Berechnung der Grundoperationen(+,-,\*,/) von zwei komplexen Zahlen sowie eine Funktion *conj*, die zu einer komplexen Zahl  $z = a + b \cdot i$  die konjugiert komplexe Zahl  $\bar{z} = a - b \cdot i$  zurückgibt.

c)

- Schreiben Sie eine Datenstruktur *Vect* für Vektoren mit beliebiger Länge über die komplexen Zahlen.
- Implementieren Sie Funktionen zur Berechnung von Addition und Multiplikation von zwei Vektoren (*Vect*) sowie eine Funktion für die Skalarmultiplikation eines Vektors (*Vect*) mit einem Skalar.