

# Übungsaufgaben 1

Arne Schmitt

15. Juli 2006

## 1 Allgemein

Für die Lösung können die Funktionen aus anderen Aufgaben verwendet werden. Bei der Namensgebung darauf achten.

Wenn keine expliziten Angaben gemacht werden wie etwas gemacht werden soll, kann man die Details selber entscheiden. Also, z.B. ob man das erste macht was einem einfällt, den einfachsten und schnellsten Weg verwendet, Lösungen von anderen Aufgaben verwendet oder versucht das Problem möglichst allgemein zu lösen um es später wiederverwenden zu können.

Diese Entscheidungen muss man in der Praxis, anderes als in der Schule dauern treffen. Deshalb fangen wir so für wie möglich damit an um Übung zu bekommen.

Bei Fragen anrufen, chatten oder Mail an [arne@asm.de](mailto:arne@asm.de)

## 2 Level 1

1. Schreibe eine Funktion die ein Dreieck aus Sternen "\*" ausgibt, dabei werden **Höhe** und **Breite** als **Parameter** angegeben.
2. Erweitere die Funktion aus der letzten Aufgabe so, daß die Spitze des Dreiecks bei positiver Höhe oben und bei negativer Höhe unten erscheint.
3. Erweitere die Funktion aus der letzten Aufgabe so, daß das Dreieck, durch einen Parameter gesteuert, entweder mit "\*" oder mit "." gefüllt wird.
4. Schreibe ein Programm das den Nutzer nach den Koordinaten zweier Punkte fragt, und das die Gleichung einer Geraden durch die beiden Punkte ausgibt.
5. Schreibe eine Funktion die ein Rechteck aus Sternen auf dem Bildschirm ausgibt.
6. Schreibe eine Funktion die ein auf einer Ecke stehendes Rechteck aus Sternen auf dem Bildschirm ausgibt.
7. Schreibe eine Funktion die eine Zeichenkette umdreht. z.B. aus "Hallo" wird "ollaH"

- Schreibe eine Funktion die zählt wie oft jeder Buchstabe in einer Zeichenkette vorkommt. z.B. "Hallo" => H=1 a=1 o=1 l=2
- Schreibe eine Funktion die ein **int array** aufsteigend sortiert.
- Schreibe eine Funktion die eine Zeichenkette sortiert. z.B. aus "Olaf" wird "afO" oder "Oaff"

### 3 Dateien

- Schreibe ein Programm das "Hallo" in die Datei "Hallo.txt" schreibt.
- Schreibe ein Programm das den Inhalt der Datei "Hallo.txt" ausgibt.
- Schreibe ein Programm das Zahlen die der Nutzer eingibt in ein Datei schreibt.
- Schreibe ein Programm das die in der letzten Aufgabe erzeugte Datei liest und die Summe und der Durchschnitt der Zahlen ausgibt.
- Schreibe ein Programm das eine Text aus einer Datei liest und zählt wie oft jeder Buchstabe vorkommt.
- Erweitere das letzte Programm so, das es das Ergebniss in eine Datei schreibt.

### 4 Strukturen

- Entwickle eine Struktur um die Position eines Punktes im dreidimensionalen Raum zu speichern.
- Schreibe eine Funktion um den Abstand von zwei Punkten zu berechnen.
- Entwickle eine Struktur um Schulnoten zu speichern.
- Entwickle eine Struktur um Telefonnummern zu speichern.
- Erweitere die Struktur der letzten Aufgabe um auch die Adresse zu verwalten.
- Erweitere die Struktur der letzten Aufgabe um mehrere Telefonnummern und EMail Adressen zu verwalten.
- Entwickle eine Struktur um eine Bestellung bis zur Lieferung und Bezahlung verwaltet.
- Schreibe ein Programm das eine Bestellung aufnimmt, den Lieferschein und die Rechnung ausgibt und den Termin des Lieferung und des Zahlungseingangs festhält.

## 5 Listen 1

### 5.1 Einfachverkettet

Wir verwenden in diesem Abschnitt folgende Definition für die Elemente unserer Liste.

```
struct Entry
{
Entry *next;
int value;
};
```

Element \*first = new Entry ();

1. Schreibe eine Funktion die den Wert des ersten Elements auf 42 setzt.
2. Schreibe eine Funktion die alle Werte nummeriert ansgibt.
3. Schreibe eine Funktion die ein neues Element am Anfang der Liste einfügt. Der Wert soll als **Parameter** übergeben werden.
4. Schreibe eine Funktion die das n-te Element ausgibt , wenn es existiert. n soll als **Parameter** übergeben werden.
5. Schreibe eine Funktion die ein neues Element am Ende der List einfügt;
6. Schreibe eine Funktion die das Element am Ende der List löscht;
7. Schreibe eine Funktion die die Zahl der Elemente in der Liste ermittelt;
8. Schreibe eine Funktion die das Element mit einem bestimmten Wert in der Liste findet.
9. Schreibe eine Funktion die einen bestimmten Wert in der Liste durch einen Anderen ersetzt.
10. Schreibe eine Funktion die ein neues Element nach einem bestimmten Wert in der Liste einfügt.
11. Schreibe eine Funktion die ein neues Element vor einem bestimmten Wert in der Liste einfügt.
12. Schreibe eine Funktion die einen bestimmten Wert in der Liste löscht.
13. Schreibe eine Funktion die einen bestimmten Wert in die Liste einträgt , wenn er noch nicht in der Liste vorkommt.
14. Schreibe eine Funktion die alle Elemente der Liste in umgekehrter Reihenfolge ausgiebt.
15. Schreibe eine Funktion die eine Kopie der Liste erzeugt.

16. Schreibe eine Funktion die eine Kopie der Liste in umgekehrter Reihenfolge erzeugt.

## 5.2 Doppeltverkettet

Wir verwenden in diesem Abschnitt folgende Definition für die Elemente unserer Liste.

```
struct Entry
{
Entry *next,*prev;
int value;
};
```

```
Element *first = new Entry ();
```

1. Schreibe eine Funktion die den Wert des ersten Elements auf 42 setzt.
2. Schreibe eine Funktion die alle Werte nummeriert ans gibt.
3. Schreibe eine Funktion die ein neues Element am Anfang der Liste einfügt. Der Wert soll als **Parameter** übergeben werden.
4. Schreibe eine Funktion die das n-te Element ausgibt , wenn es existiert. n soll als **Parameter** übergeben werden.
5. Schreibe eine Funktion die ein neues Element am Ende der List einfügt;
6. Schreibe eine Funktion die das Element am Ende der List löscht;
7. Schreibe eine Funktion die die Zahl der Elemente in der Liste ermittelt;
8. Schreibe eine Funktion die das Element mit einem bestimmten Wert in der Liste findet.
9. Schreibe eine Funktion die einen bestimmten Wert in der Liste durch einen Anderen ersetzt.
10. Schreibe eine Funktion die ein neues Element nach einem bestimmten Wert in der Liste einfügt.
11. Schreibe eine Funktion die ein neues Element vor einem bestimmten Wert in der Liste einfügt.
12. Schreibe eine Funktion die einen bestimmten Wert in der Liste löscht.
13. Schreibe eine Funktion die einen bestimmten Wert in die Liste einträgt , wenn er noch nicht in der Liste vorkommt.
14. Schreibe eine Funktion die alle Elemente der Liste in umgekehrter Reihenfolge ausgibt.

15. Schreibe eine Funktion die eine Kopie der Liste erzeugt.
16. Schreibe eine Funktion die eine Kopie der Liste in umgekehrter Reihenfolge erzeugt.