

Du bist immer noch nicht hemmungslos überfordert? Dann $f^* + / - n$ wir dich jetzt so richtig! Versuche doch mal folgende 3 Aufgaben in einer halben Stunde zu lösen.

Aufgabe 1. In dieser Aufgabe sollst du eine Datenbank von Studentinnen programmieren. Dazu verwenden wir doppelt verkettete Listen.

- a) Schreibe ein struct `studentin`, welches Einträge für Name, Vorname, Matrikelnummer, Notendurchschnitt, Fachsemester und ein Array der Länge 20 mit den Noten aus all ihren Prüfungen enthält. Wenn du magst, kannst du dir natürlich weitere Einträge überlegen.
- b) Schreibe Funktionen um einige oder alle diese Werte für eine gegebene `studentin` zu setzen und schreibe eine Funktion, um die Werte einer `studentin` auf der Konsole auszuprinten. Achte dabei auf eine schöne Formatierung.
- c) Schreibe eine doppelt verkettete Liste, welche als Daten Pointer auf den struct `studentin` speichern kann.
- d) Schreibe eine Funktion, welche Speicher für eine neue `studentin` allokiert und einen Pointer auf diesen zurückgibt. Fülle diese `studentin` mit deinen (Wunsch-)Daten und ~~schicke diese an Google~~ füge diesen Eintrag der Liste hinzu.
- e) Wiederhole den vorherigen Schritt mit den Daten deiner Freunde. Wenn du keine Freunde hast, denke dir welche aus.
- f) Schreibe eine Funktion, um dir den gesamten Inhalt der Liste auf der Konsole auszugeben.
- g) Eine `studentin` wird exmatrikuliert und muss deshalb aus der Liste gelöscht werden.
- h) Sorge dafür, dass alle Elemente wieder einwandfrei gelöscht werden und gebe allen Speicher frei, den du benutzt hast.

Aufgabe 2. Schreibe ein Programm, das ein Labyrinth aus einer Datei einliest:

```
XXXXXXXXXXXXXXXXXX
X X XXXXXXXXXXXX*X
X$X XX      XXX X
X X XX XXX XXX X
X   XX XXX XXX X
XXX X   XX XXX X
XXX  X           X
XXXXXXXXXXXXXXXXXX
```

Bemerkung: Wir spezifizieren das Labyrinth hier nicht viel näher, entscheide dich selbst vorher was für ein Format die Datei haben soll und welche Einschränkungen du daran stellst: Soll die Größe des Labyrinths variabel sein oder fest? Soll die Größe in der ersten Zeile der Datei stehen oder nicht? Soll das Labyrinth quadratisch sein oder nicht? Soll es außen herum immer mit Xen begrenzt sein oder hast du vielleicht eine andere Lösung?

Das Programm soll einen Weg vom Startpunkt (dem Stern, dem Geburtsort) zum Dollar (dem Schatz) finden. Die Xe sind Wände und Leerzeichen sind Pfade. Markiere einen Weg mit Punkten und gebe das Labyrinth mit Weg in der Konsole aus.

```
XXXXXXXXXXXXXXXXXX
X X XXXXXXXXXXXX*X
X$X XX      XXX.X
X.X XX XXX XXX. X
X...XX XXX XXX. X
XXX.X...XX XXX.X
XXX...X.....X
XXXXXXXXXXXXXXXXXX
```

Aufgabe 3. Brainfuck ist eine sogenannte esoterische Programmiersprache, das sind Sprachen, die meist zu wissenschaftlichen oder theoretischen Zwecken, oder einfach zum Spaß entwickelt wurden.

Brainfuck besteht nur aus 8 Befehlen: > < + - , . [] alle anderen Zeichen werden als Kommentar interpretiert. Diese Befehle werden, wie bei C auch, nacheinander ausgeführt. Sie operieren auf einem (potentiell unendlich langen) Band (welches aus Zellen besteht in denen jeweils ein char steht)

indem sie einen Lese-/Schreibkopf über das Band bewegen und Zeichen lesen / schreiben lassen. Das Band ist überall mit '\0' vorinitialisiert und der Lese-/Schreibkopf startet an "Position 0" des Bandes. Die Befehle haben folgende Bedeutung:

> bzw. <	schiebt den Lese-/Schreibkopf eins nach rechts bzw. links
+ bzw. -	in- bzw. dekrementiert den Bandwert unter dem Lese-/Schreibkopf um 1
.	gibt den Wert unter dem Lese-/Schreibkopf aus
,	liest ein Zeichen vom Benutzer ein und schreibt es unter den Lese-/Schreibkopf
[springt zum zugehörigen]-Befehl, wenn der Wert unter dem Lese-/Schreibkopf 0 ist, sonst soll nichts passieren
]	springt zum zugehörigen [-Befehl, wenn der Wert unter dem Lese-/Schreibkopf verschieden von 0 ist

So sieht ein "Hallo-Welt"-Programm in Brainfuck aus:

```
1 ++++++
2 [
3   >+++++>+++++>+++>+<<<<-
4 ]
5 >++.
6 >+.
7 ++++++.
8 +++.>+.
9 <<+++++.
10 >.++.
11 _____ .
12 >+.>.
```

Deine Aufgabe ist es nun, ein Programm zu schreiben, welches Brainfuck-Programme einlesen und ausführen kann.