

1) Struct/Exceptions: Schreibe eine struct Knoten für eine SLL und Integerwerten als Daten. Im Hauptprogramm definiere daraus direkt, d.h. ohne Funktion, eine Liste mit einem Anker und mind. 5 Knoten.

Dann schreibe die Funktion **finde**, welche einen bestimmten Wert in der Liste sucht und die Position wo er zuletzt gefunden wurde retourniert (erstes Element ist dabei 0, zweites 1, ...) – gibt es den Wert nicht soll eine passende Ausnahme geworfen werden.

Definieren sich auch eine rekursive Version für das **abbauen** der Liste und verwenden sie!

2) Klasse/Modularisierung: Schreiben sie eine Klasse Lehrer welche folgenden Eigenschaften besitzen soll: `string* name` (verpflichtend, nicht änderbar), `int spassfaktor` (änderbar). Die Anzahl der aktuellen Lehrerinstanzen soll dabei zu jeder Zeit über entsprechende Funktion abfragbar sein!

Implementieren sie mittels `.h` und `.cpp` Datei einen passenden Dekonstruktor und Konstruktor sowie die sonstigen Zugriffsmethoden.

3) Rekursive Sort: Schreibe eine Funktion `rec_selection_sort` für ein `int`-Array, folgender Pseudo-Code ist gegeben:

- 1) Finde das kleinste Element in der Eingabe-Liste.
- 2) Tausche dieses Element mit erstem (dann zweiten, dritten, ...) Element in der Liste
- 3) wenn die Eingabe-Liste noch nicht leer ist, gehe zu Schritt 1) zurück.

TIP: Sie können auch eine rekursive Hilfsfunktion zum Finden des kleinsten Elementes schreiben!

4) dyn. Arrays: Schreiben sie eine Funktion

```
int* StrToInt(string ein, int &anz)
```

wo sie aus einem Text die Zahlen als dynamisches `Int`-Array zurückgeben und wiederum im Hauptprogramm ausgeben lassen

5) Kommandozeilenparameter/Dateiverarbeitung: Schreiben sie ein Programm welches über die Kommandozeile einen Dateinamen erwartet und mittels einer Funktion versucht diese zu öffnen und nur die Anzahl der darin enthaltenen Zahlen (0-9) zurückgeben werden soll. Ausnahmenbehandlung!