



Lernzielkontrolle: Datenstrukturen

Name: _____ Datum: _____

- Was ist der Vorteil eines dynamischen Arrays?
·
- Was ist der Nachteil eines dynamischen Arrays?
·
- Nennen Sie weitere Datenstrukturtypen, welche nicht diesen Nachteil besitzen:
·
·
- **Programmieraufgabe D1:** Speichern Sie die Personen (P sei der Typ) der vorhergehenden Aufgabe in einem dynamisch wachsenden Strukturarray, das modullokal gespeichert wird. Verwenden Sie dazu folgende Strategie:
 - Speichern Sie zuerst nur einen modullokalen Zeiger $P *p$, der auf NULL zeigt und merken Sie sich ebenfalls modullokal die Anzahl der gespeicherten Personen n , die anfangs 0 ist.
 - Schreiben Sie eine Funktion „add_person($P *p$)“, die eine neue Person p einfügt und diese via CBR als Strukturparameter geliefert bekommt. Speichern Sie diese in einem neuen dynamisch allozierten lokalen Array der Größe $n+1$ am Index n .
 - Die Stellen 0 bis $n-1$ des vorhergehenden Arrays der Größe n werden herüberkopiert. Danach wird das alte Array gelöscht und der modullokale Zeiger auf das neue Array gesetzt und n erhöht.
 - Ist dieses Vorgehen effizient? Welche entscheidende Verbesserung wäre mit wenig Aufwand realisierbar?
 - Testen Sie Ihr Programm, indem Sie alle Personen aus der vorhergehenden Übung und sich selber einfügen.
 - Wieviele Elemente hat das Array nun? → get_num_persons()
- **Programmieraufgabe D2:** Schreiben Sie eine Suchfunktion „search_person(char *v, char *n)“, die eine bestimmte Person anhand des Vornames v und des Nachnamens n sucht.
 - Die Funktion soll alle Personen der Reihe nach durchgehen und auf Namensgleichheit überprüfen → `strcasecmp(a, b) == 0`.
 - Die Suchfunktion soll einen Zeiger auf die erste namensgleiche Person zurückgeben, ansonsten den NULL Zeiger.
 - Testen Sie Ihr Programm, indem Sie Ihr eigenes Geburtsdatum ausgeben! Geben Sie in main() Ihren Geburtsmonat zurück. Laden Sie das Ergebnis in SVN im Ordner „search“ hoch.