

Arbeitsblatt: INF1

Name:

Kurznamen:

Speicherverwaltung, Arrays.

Aufgabe 1: Datenstrukturen mit dynamischen Arrays

Im letzten Praktikum haben wir eine Rangliste erstellt. Dabei muss aber die maximale Anzahl Teilnehmer als Konstante im Programm festgelegt werden. Wesentlich flexibler ist es, wenn die Anzahl der Teilnehmer als Array dynamisch (i.e. zur Laufzeit) festgelegt werden kann. Die Teilnehmerzahl wird einfach als zusätzliche erste Eingabe abgefragt. Die Eingabe soll beendet werden, sobald die vorgegebene Anzahl der Teilnehmer erreicht ist.

Hinweise:

- Nehmen Sie als Basis die Lösung *ohne den Comparator*
- Passen Sie das Ranglisten Programm so an, dass die Anzahl zur Laufzeit angegeben werden kann und verwenden Sie einen dynamisch allozierten Array.
- Die Freigabe können Sie sich in diesem Beispiel sparen, da bei Programmende sowieso aller Speicher wieder freigegeben wird.

Abgabe

Praktikum: INF10.1

Filename: rangliste.c

Aufgabe 2: Datenstruktur mit dynamischen Daten (struct)

Jetzt wollen wir einen Schritt weiter gehen und die Teilnehmer dynamisch allozieren. Das heisst, auch das Teilnehmer Struct wird erst zu Laufzeit erzeugt und es wird jeweils mit Zeigern auf die Teilnehmer gearbeitet. Schreiben Sie das Programm so um, dass konsequent (in allen Funktionen) der Pointer auf die Teilnehmer verwendet wird.

Hinweise:

- Definieren Sie einen neuen Typ: `typedef Teilnehmer* TeilnehmerPtr;`
- Ersetzen Sie konsequent Teilnehmer durch TeilnehmerPtr in Ihrem Programm
- Nehmen Sie die Vorlage, z.B

```
TeilnehmerPtr leseTeilnehmer (void) {
    TeilnehmerPtr pt = (TeilnehmerPtr) calloc(1, sizeof(Teilnehmer));
    printf("Name: ");
    scanf(" %19[^\n]s", pt->name);
    printf("Zeit (h:min:sec): ");
    scanf(" %d:%d:%d", &(pt->h), &(pt->min), &(pt->sec));
    return pt;
}
```

Abgabe

Praktikum: INF10.2

Filename: rangliste.c