

Praktikum zu "Grundlagen der Programmierung"**Testat 4****Vorführaufgabe 4**

Viele informationstechnische Probleme erfordern die Speicherung von Informationen und ein schnelles Wiederauffinden bestimmter, gegebener Information.

Wenn nun die Information mittels eines numerischen Identifikators eindeutig identifiziert werden kann, so bietet sich eine Reihung an. Mit dem Identifikator als Index kann direkt auf die Information zugegriffen werden.

Beispiel:

Studierende werden über ihre fünfstellige Matrikelnummer eindeutig identifiziert.

Sie sollen ein einigermaßen zukunftsträchtiges Studentenverwaltungssystem programmieren. Wie groß würden Sie ein Feld `students` mit Zeigern auf Strukturen vom Typ `struct Student` auslegen? Wie viele Einträge werden wohl belegt sein?

Eine Alternative besteht in einer sogenannten Hash-Tabelle. Anstatt für jede irgendwann einmal verwendete Matrikelnummer ein Feldelement vorzusehen, wird ein deutlich kleineres Feld verwendet (z.B. nur 20 Elemente). Um zu einer gegebenen Matrikelnummer `mat_nr` den Index des Feldelementes, in dem weitere Angaben über den Speicherort der gesuchten Information gespeichert sind, zu finden, wird eine so genannte Hash-Funktion h verwendet, z.B.

$$h(\text{mat_nr}) = \text{mat_nr} \% 20$$

Die Hash-Funktion liefert zu einer Matrikelnummer einen Index, mit dem dann auf das Feld zugegriffen wird. Dort findet sich dann eine Referenz auf die gesuchte Information (hier: die Struktur, welche den Studenten mit der Matrikelnummer `mat_nr` repräsentiert).

Bei Verwendung einer Hash-Funktion kann es aber passieren, dass zwei unterschiedliche Identifikationen (hier: Matrikelnummern) auf den gleichen Index abgebildet werden – es kommt zu einer sogenannten Kollision. Eine Möglichkeit, diese Kollisionen zu behandeln, besteht darin, statt eines einzelnen Elements eine Liste von Elementen zu referenzieren. Diese Liste muss dann nach dem gesuchten Identifikator durchsucht werden.

- a. Definieren Sie Strukturen für
 - die Repräsentierung eines Studierenden (`struct Student` mit Matrikelnummer und Name), die eine Verkettung zulässt
 - ein Feld mit 20 Zeigern auf Strukturen vom Typ `struct Student`
- b. Implementieren Sie eine Funktion `hash`, welche die Hash-Funktion berechnet.
- c. Implementieren Sie eine Prozedur `insert_student`, die unter Berücksichtigung von Kollisionen eine neue `Student`-Struktur erzeugt und in die Datenstruktur einfügt. Parameter der Prozedur sind die Matrikelnummer und der Name des einzufügenden Studenten.
- d. Implementieren Sie eine Prozedur `print_student`, die zu gegebener Matrikelnummer den Namen des Studenten ausgibt.
- e. Implementieren Sie eine `main`-Funktion, in der Sie so lange Matrikelnummer und Name von Studenten eingeben und in die Hash-Tabelle einfügen, bis eine 0 als Matrikelnummer eingegeben wird. Dann lassen Sie sich so lange den Namen eines Studenten zu einer eingegebenen Matrikelnummer ausgeben, bis wieder eine 0 eingegeben wird.