

Kennzahl: _____

Herbst

Kennwort: _____

1998

46111

Arbeitsplatz-Nr.: _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
- Prüfungsaufgaben -

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Programmentw./Systempr./Datenbanksys.**

Anzahl der gestellten Themen (Aufgaben): 1

Anzahl der Druckseiten dieser Vorlage: 4

Bitte wenden!

Sämtliche Teilaufgaben sind zu bearbeiten!**Aufgabe 1 (Programmentwicklung)**

Der folgende Datentyp CharList wird zur Darstellung von Worten (Strings) mit maximal 1000 Buchstaben verwendet. Für jedes Wort w des Typs CharList gibt $w.laenge$ die Länge des Wortes an. Die einzelnen Buchstaben von w sind in den Feldkomponenten $w.elemente[1]$, ..., $w.elemente[w.laenge]$ gespeichert. (In den Feldkomponenten $w.elemente[i]$ mit $i > w.laenge$ können beliebige Zeichen gespeichert sein.) Für das leere Wort gilt $w.laenge = 0$.

```
TYPE  Laenge = [0..1000];
      Index = [1..1000];
      CharList = RECORD
          laenge: Laenge;
          eiemente: ARRAY Index OF CHAR
      END;
```

In der Teilaufgabe a) sollen einige Standardprozeduren für Worte erstellt werden. In der Teilaufgabe b) sollen diese Prozeduren zur Lösung eines speziellen Wort-Problems verwendet werden. In Teilaufgabe c) ist die (Ordnung der) Zeitkomplexität aller Prozeduren zu bestimmen. Die Prozeduren sind in der Syntax von Modula oder von Pascal zu schreiben. Die Prozedurköpfe sind im folgenden jeweils in der Syntax von Modula angegeben.

a) Zu erstellen sind folgende Prozeduren:

1. Eine Funktionsprozedur mit Kopf `PROCEDURE length (w: CharList) : Laenge;` die die Länge eines Wortes w liefert.
2. Eine Funktionsprozedur mit Kopf `PROCEDURE get (w: CharList; i: Index) : CHAR;` die das i -te Element eines Wortes w liefert. (Es kann davon ausgegangen werden, daß w nicht leer ist und daß $i \leq \text{length}(w)$ gilt.)
3. Eine reine Prozedur mit Kopf `PROCEDURE insert (VAR w: CharList; c: CHAR; i: Index);` sodaß nach Ablauf der Prozedur das Zeichen c an der i -ten Stelle in das Wort w eingefügt ist (bzw. angehängt ist, falls $i = \text{length}(w)+1$). (Es kann davon ausgegangen werden, daß $i \leq \text{length}(w)+1 \leq 1000$ gilt.)
4. Eine reine Prozedur mit Kopf `PROCEDURE delete (VAR w: CharList; i: Index);` ~~sodaß~~ **sodaß nach Ablauf der Prozedur das i -te Zeichen aus dem Wort w entfernt ist.** (Es kann davon ausgegangen werden, daß w nicht leer ist und daß $i \leq \text{length}(w)$ gilt.)

b) Ein Wort w ist ein "Spiegelwort" genau dann, wenn es von vorne und von hinten gelesen dieselbe Zeichenkette bildet (z. B. OTTO).

1. Schreiben Sie eine nicht-rekursive Funktionsprozedur mit Kopf `PROCEDURE ist_spiegelwort (w: CharList) : BOOLEAN;` die feststellt, ob w ein Spiegelwort ist!
2. Schreiben Sie eine rekursive Funktionsprozedur mit Kopf `PROCEDURE ist_spiegelwort_rek (w: CharList) : BOOLEAN;` die feststellt, ob w ein Spiegelwort ist!

Fortsetzung nächste Seite!

- c) Bestimmen Sie für jede der in Teil a) und b) erstellten Prozeduren die Größenordnung der Zeitkomplexität im schlechtesten Fall ("worst case") in Abhängigkeit von der Länge n des Wortes w .

Aufgabe 2 (Systemprogrammierung)

Im folgenden sei ein Rechner mit einem Prozessor und einem Mehrprogramm-Betriebssystem (zur Verwaltung mehrerer parallel ablaufender Prozesse) gegeben.

- a) Geben Sie die verschiedenen Zustände an, in denen sich ein Prozeß befinden kann! Erläutern Sie, welche Übergänge zwischen den einzelnen Zuständen eines Prozesses möglich sind, und nennen Sie jeweils eine Ursache für einen Zustandswechsel!
- b) Beschreiben Sie ein Zuteilungsverfahren, mit dem der Prozessor den einzelnen Prozessen zugeteilt werden kann!
- c) Gegeben seien die folgenden Prozeßbeschreibungen zweier parallel ablaufender Prozesse P_1 und P_2 . Jeder Prozeß P_i ($i = 1, 2$) führt in einer nichtterminierenden WHILE-Schleife zunächst eine Anweisungsfolge A_i aus und möchte danach auf ein gemeinsames Betriebsmittel B zugreifen.

```
P1:  WHILE TRUE DO
      A1 ;
      Zugriff auf B
    END;
```

```
P2:  WHILE TRUE DO
      A2 ;
      Zugriff auf B
    END;
```

Die beiden folgenden Teilaufgaben c1) und c2) sind unabhängig voneinander zu bearbeiten!

- c1) Beschreiben Sie, wie man unter Verwendung eines Semaphors die beiden Prozesse so synchronisieren kann, daß sie nicht gleichzeitig auf das Betriebsmittel B zugreifen können, jedoch ihre Anweisungsfolgen A_1 und A_2 parallel ausführen können!
- c2) Die beiden Prozesse sollen so synchronisiert werden, daß sie abwechselnd auf das Betriebsmittel B zugreifen. Geben Sie eine Lösung dieses Problems mit Hilfe von zwei Semaphoren an, bei der Prozeß P_1 als erster auf das Betriebsmittel B zugreift! (Vergessen Sie dabei nicht, die Semaphore geeignet zu initialisieren!)

Aufgabe 3 (Datenbanksysteme)

Eine Bibliothek möchte eine Datenbank anlegen. Ausgangspunkt sind eine Relation "Buch" (für die Bücher der Bibliothek) und eine Relation "Ausleihe" (mit Angaben zu den aktuell ausgeliehenen Büchern). Die Relation "Buch" hat die Attribute

Buch-Nr Titel Autor Verlag

und die Relation "Ausleihe" hat die Attribute

Buch-Nr Entleiher-Nr Ausleihdatum Entleiher-Name Entleiher-Telefon

wobei "Buch-Nr" der Schlüssel jeder Relation ist.

- a) Die Relation "Ausleihe" ist in zweiter Normalform (2NF) aber nicht in dritter Normalform (3NF). Geben Sie jeweils eine Begründung dafür an!
- b) Geben Sie drei Nachteile an, die sich bei der Verwendung der Relation "Ausleihe" ergeben können!
- c) Überführen Sie die Relation "Ausleihe" in zwei Relationen in dritter Normalform, und geben Sie den beiden neuen Relationen jeweils einen Namen!

Die Bibliotheksdatenbank soll nun die Relation "Buch" und die beiden in Teil c) konstruierten Relationen enthalten. Formulieren Sie folgende Anfragen an die Datenbank in SQL:

- d) Finde die Titel aller Bücher, die vom Autor "Wirth" verfaßt wurden.
- e) Finde für jedes ausgeliehene Buch die Buch-Nr, den Titel, den Autor und die Entleiher-Nr.
- f) Finde den Titel und den Autor aller nicht ausgeliehenen Bücher.
- g) Finde den Namen und die Telefonnummer jedes Entleihers, der mehr als 10 Bücher ausgeliehen hat.