Vordiplom Wirtschaftswissenschaften
Allgemeine Informatik I
WS 2000/2001
21. Februar 2001

Bearbeitungszeit: 120 Minuten

Name:	
Vorname:	
Matrikelnummer:	

Aufgabe	Punkte	Bewertung
1	**18**	
a)	6	
b)	6	
c)	6	
2	**18**	
a)	2	
b)	2	
c)	4	
d)	2	
e)	4	
f)	4	
3	**14**	
a)	4	
b)	6	
c)	4	
4	**6**	
5	**8**	
6	**10**	
7	**6**	
8	**10**	
Summe:	**90**	
	Note:	

Bitte benutzen Sie für die Lösungen den freigelassenen Platz nach der jeweiligen Angabe; sollte dieser nicht ausreichen, verwenden Sie bitte die Rückseite, wobei die Zuordnung zur jeweiligen Aufgabe deutlich erkennbar sein muss!
Viel Erfolg!!!

Aufgabe 1 **(18 Punkte)**

Die formale Sprache L über dem Vokabular $V = \{0, 1\}$ sei verbal wie folgt definiert:
$L = \{$ alle mit '1' beginnenden, nichtleeren Zeichenfolgen über V, bei denen die Anzahl der unmittelbar aufeinander folgenden Einsen '1' gerade und die Anzahl der unmittelbar aufeinander folgenden Nullen '0' ungerade ist $\}$

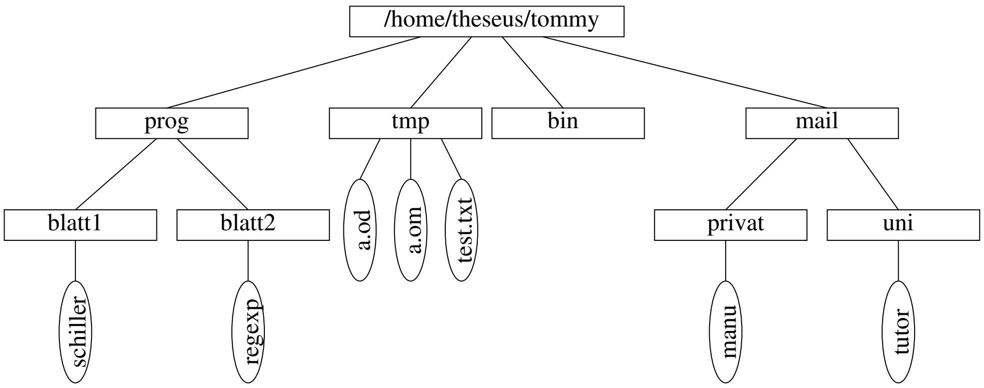
a) 6 Punkte
Beschreiben Sie L mittels einer regulären Grammatik.

b) 6 Punkte
Beschreiben Sie L mittels eines regulären Ausdrucks.

c) 6 Punkte
Beschreiben Sie L mittels eines deterministischen endlichen Automatens (in grafischer Darstellung).

Aufgabe 2 **(18 Punkte)**

Gegeben ist im Folgenden die grafische Darstellung eines Auszugs aus der Verzeichnisstruktur des Benutzers *tommy*, der ebenfalls an der Vorlesung Allgemeine Informatik I für WiWis teilnimmt. (Die Rechtecke stellen Katalogdateien und die Ellipsen reguläre (gewöhnliche) Dateien dar):



a) 2 Punkte
tommy hat sich angemeldet und befindet sich nun in seinem Heimatkatalog. Da er nun das 3. Übungsblatt beginnen will, legt er zunächst den Unterkatalog *blatt3* im Katalog *prog* an, ohne dabei den aktuellen Katalog */home/theseus/tommy* zu wechseln. Geben Sie das notwendige Kommando an!

`theseus$` ______________________________

b) 2 Punkte
Gestern hat *tommy* schon an einer Aufgabe von Blatt 3 gearbeitet und dafür die Module *a.om* und *a.od* geschrieben. Diese liegen momentan im Katalog *tmp*. Er möchte nun, ohne seinen aktuellen Katalog zu wechseln (er befindet sich noch immer im Heimatkatalog), dieses Oberon-Programm in den Katalog, welchen er für Blatt 3 angelegt hat, verschieben. Geben Sie das notwendige Kommando an!

`theseus$` ______________________________

c) 4 Punkte
Nun wechselt er vom Heimatkatalog in den Katalog für Blatt 3 und möchte aus diesen beiden Modulen eine ausführbare Datei erzeugen. Die beim Erzeugen der ausführbaren Datei auftretenden Fehlermeldungen des Compilers will er in die Datei *errors* umleiten. Geben Sie die hierzu erforderlichen Kommandos an! Die

Anzahl der notwendigen Kommandos muss nicht mit der folgenden Anzahl von Kommandozeilen identisch sein.

theseus$ ______________________________

theseus$ ______________________________

theseus$ ______________________________

d) 2 Punkte
tommy sieht anhand der Fehlermeldungen in *errors*, dass er eine Prozedur aus dem Modul *Read* verwendet hat, die es gar nicht gibt. Er möchte also wissen, welche Prozeduren in diesem Modul enthalten sind. Was kann er tun, wenn er am Rechner sitzt, das Skript nicht zur Hand hat und momentan niemand zu sehen ist, den er fragen könnte (ein Telefon hat er auch nicht)?

e) 4 Punkte
Nach dem Verbessern der Fehler möchte er nun endlich das ausführbare Programm erzeugen (was diesmal klappen wird). Danach kopiert er die ausführbare Datei in den Katalog *bin* und löscht alle vom Compiler generierten Dateien. Geben Sie die erforderlichen Kommandos an, wobei die Anzahl der folgenden Zeilen nichts über die Anzahl der notwendigen Kommandos aussagt. (Aktueller Katalog ist *blatt3*.)

theseus$ ______________________________

theseus$ ______________________________

theseus$ ______________________________

f) 4 Punkte
Nun möchte er das ausführbare Programm starten, wobei die Standardeingabe auf die Datei *test.txt* aus dem Katalog *tmp* und die Standardausgabe in die Datei *ausgabe* ebenfalls im Katalog *tmp* umgelenkt werden soll. Momentan befindet sich *tommy* im Katalog *blatt3* und will diesen nicht wechseln. Geben Sie das notwendige Kommando an!

theseus$ ______________________________

Aufgabe 3 **(14 Punkte)**

Die Datei *druckseiten* enthält eine Datenbank der Druckkontingente aller Personen mit einem Login auf den Rechnern in unserer Fakultät. Jede Zeile der Datei entspricht einem Datensatz. Jeder Datensatz besteht wiederum aus den Feldern Login, verbleibendes Druckkontingent und Datum des letzten Ausdrucks. Diese sind voneinander durch ':' getrennt. Das Druckkontingent ist eine ganze Zahl. Das Datum liegt im Format *Tag* '.' *Monat* '.' *Jahr* vor.
Beispiel (einige Zeilen aus der Datei *druckseiten*):

```
susi:142:28.04.1998
franz:200:17.6.99
tommy:017:20.8.2000
```

Sie können davon ausgehen, dass die Datei *druckseiten* genau das beschriebene Format besitzt (und in keinem Feldwert das Zeichen ':' vorkommt). Die Felder Druckkontingent und Datum können auch leer sein.

a) 4 Punkte
Beschreiben Sie in Worten, was die folgende Kommandozeile tut:
`cat druckseiten | egrep '::' | wc -l`

b) 6 Punkte
Geben Sie eine Kommandozeile an, die alle Datensätze der Datei *druckseiten* auf die Standardausgabe ausgibt, bei denen das Druckkontingent mindestens 100 Seiten umfasst. (Eine ganze Zahl kann auch führende Nullen besitzen!)

`theseus$` ____

c) 4 Punkte
Am Ende des vorigen Jahrtausends war das Wort „Jahr-2000-Problem“ in aller Munde. Hintergrund war die Tatsache, dass von einigen Programmen die Jahresangabe im Datum (ursprünglich mangels Speicherplatzes) nur mit zwei Stellen abgespeichert wurde (d. h. ohne Jahrtausend und ohne Jahrhundert). In der Datei *druckseiten* gibt es offensichtlich ebenfalls solche Datumsangaben. Schreiben Sie ein Kommando, das alle Zeilen der Datei *druckseiten* ausgibt, die ein Datum mit zweistelliger Jahreszahl enthalten.

`theseus$` ____

Aufgabe 4 **(6 Punkte)**

Definieren Sie eine Oberon-Datenstruktur, um einen Druckkontingent-Datensatz wie in Aufgabe 3, d. h. Login, Druckkontingent und Datum des letzten Ausdrucks, zu speichern. Verwenden Sie hierzu Konstanten- und Typdefinitionen. Der Login-Name besteht aus maximal 8 Zeichen. Das Druckkontingent ist eine ganze Zahl. Tag, Monat und Jahr des Datums sollen über die Datenstruktur direkt („einstufig“) zugreifbar sein.

Aufgabe 5 **(8 Punkte)**

Der Chef-*Controller* eines Konzerns benötigt für die einzelnen Tochter-Gesellschaften, deren Anzahl bekannt und fest ist, die Umsatzzahlen für eine fest vorgegebene Anzahl zurückliegender Jahre.
Dazu hat er sich eine Datenstruktur überlegt, die in Oberon wie folgt aussieht:

```
CONST
  MaxJahre     = 5;
  AnzahlFirmen = 12;

TYPE
  Umsatz =        RECORD
                    jahr: INTEGER;
                    umsatz: REAL;
                  END;

  UmsatzTabelle = RECORD
                    anzahl: INTEGER;
                   (* anzahl: Anzahl der vorliegenden
                    * Jahresumsaetze
                    *)
                    jahresUmsatz: ARRAY MaxJahre OF Umsatz
                  END;

  GesamtTabelle = ARRAY AnzahlFirmen OF UmsatzTabelle;
```

Es ist davon auszugehen, dass für alle *AnzahlFirmen* Tochtergesellschaften Daten vorliegen, dass aber nicht bei jeder davon genau *MaxJahre* Jahresumsätze bekannt sind. Die Komponente *anzahl* in *UmsatzTabelle* gibt also an, wieviele Elemente (von 0 an!) in der Komponente *jahresUmsatz* tatsächlich belegt sind.
Ihre Aufgabe:
Schreiben Sie eine Prozedur, die als Parameter ein Objekt vom Typ *GesamtTabelle* sowie eine Firmennummer (im Bereich 0..AnzahlFirmen-1) erhält und die zugehörigen Umsatzzahlen in der Form **jahr:umsatz** an die Standardausgabe schreibt.
(Bitte benutzen Sie für Ihre Lösung die folgende Leerseite!)

Lösung:

Aufgabe 6 **(10 Punkte)**

Schreiben Sie eine Oberon-Prozedur **ToCapital**, die als Parameter eine Folge von Zeichen bekommt und die darin enthaltenen Klein-Buchstaben in Groß-Buchstaben umwandelt. Die so erzeugte Zeichenfolge soll dem Aufrufer der Prozedur zurückgeliefert werden.

Rahmen:

```
MODULE MyString;
  IMPORT Write;
  CONST
       Dim = 30;
  TYPE
       CharSequence = ARRAY Dim OF CHAR;
  VAR
       w: CharSequence;
(* HIER SOLLTE IHRE PROZEDUR STEHEN: *)
  PROCEDURE ToCapital(

  END ToCapital;
(*DER PLATZ BIS HIER MUESSTE GEREICHT HABEN *)
BEGIN
  w := "Franz Schweiggert!";
  ToCapital(w);
  Write.String(w); Write.Ln
END MyString.
```

Aufgabe 7 **(6 Punkte)**

Gegeben ist das folgende Programm:

```
MODULE Obscure;

   IMPORT Write, Conclusions;

   VAR a, b, c : INTEGER;

   PROCEDURE Obscure(VAR x : INTEGER; y : INTEGER);
   BEGIN
      y := x;
      x := b;
      b := y;
   END Obscure;

BEGIN
   Write.String(" a | b | c "); Write.Ln;
   Write.String("---+---+---"); Write.Ln;

   a := 3; b := 7; c := 5;

   Write.Char(" ");
   Write.Int(a, 0); Write.String(" | ");
   Write.Int(b, 0); Write.String(" | ");
   Write.Int(c, 0); Write.Ln;

   Obscure(a, b);

   Write.Char(" ");
   Write.Int(a, 0); Write.String(" | ");
   Write.Int(b, 0); Write.String(" | ");
   Write.Int(c, 0); Write.Ln;

   Obscure(c, a);

   Write.Char(" ");
   Write.Int(a, 0); Write.String(" | ");
   Write.Int(b, 0); Write.String(" | ");
   Write.Int(c, 0); Write.Ln;
END Obscure.
```

Tragen Sie in folgender Tabelle die vom Programm erzeugten Ausgaben für die Werte der Variablen a, b und c ein.

a	b	c

Aufgabe 8 **(10 Punkte)**

Schreiben Sie die folgende Oberon-Funktionsprozedur

```
PROCEDURE BinToDec(VAR dec : INTEGER; bin : ARRAY OF CHAR) : BOOLEAN;
```

Diese wandelt die im String *bin* übergebene Dualzahl (= Binärzahl) in eine Dezimalzahl um und gibt diese in der Variablen *dec* zurück. Der Rückgabewert der Funktionsprozedur soll genau dann *TRUE* sein, wenn der String *bin* nur aus Nullen ('0') und Einsen ('1') besteht. (Andernfalls sei der Rückgabewert *FALSE*; in diesem Fall ist der Inhalt der Variablen *dec* nicht von Bedeutung.) Der String *bin* muss nicht durch ein Null-Byte terminiert sein. In diesem Fall bilden alle Zeichen des Arrays *bin* den String. Sie können davon ausgehen, dass das Ergebnis der Konvertierung in eine *INTEGER*-Variable passt.