Klausur zu Programmieren I

WS 96 / 97

20. Januar 1996

Name _____

Vorname _____

Matrikel-Nr. _____

Aufgabe		Punkte	Bewertung
1	a)	4	
	b)	4	
	c)	4	
	d)	4	
	e)	6	
2		20	
3		10	
4	a)	4	
	b)	4	
	c)	6	
5		10	
6	a)	6	
	b)	6	
7		12	
Summe		**100**	

Hinweis: Bitte benützen Sie den jeweils vorgesehenen Platz. Sollte dieser nicht ausreichen, verwenden Sie bitte die Rückseite und machen Sie eine eindeutige Markierung!

Aufgabe 1

(*a*) 4 Punkte

Stellen Sie möglichst einfach fest, wieviel Unterkataloge in Ihrem Heimatkatalog sind. Sie können **nicht** davon ausgehen, daß Ihr momentaner Arbeitskatalog der Heimatkatalog ist.
Geben Sie eine passende UNIX-Befehlszeile an.

(*b*) 4 Punkte

Sie haben ein Modula-2-Programm mit dem *vi* editiert. Nun stellen Sie fest, daß Sie statt der ganzzahligen Division (DIV-Operator) immer die reelle Division (*/*) verwendet haben. Da das Zeichen "/" wirklich nur im Zusammenhang mit ganzzahliger Division vorkommt, können Sie automatisch alle Schrägstriche durch "DIV" ersetzen.
Geben Sie das Editor-Kommando an!

(*c*) 4 Punkte

Geben Sie ein UNIX-Kommando an, das die Anzahl der Leerzeilen in der Datei *text* ausgibt. Leerzeilen (in einem Text) sind Zeilen, in denen "nichts" oder eine Folge von Blanks / Tab's enthalten sind.

(*d*) 4 Punkte

Sie haben in Ihrem Heimatkatalog eine ganze Reihe von Modula-2-Quellprogrammen (also viele **.m2**-Dateien). Leider sind die Modul-/Dateinamen sehr wenig aussagekräftig, zum Glück haben Sie aber am Modulanfang aussagekräftige Kommentare. Sie wollen feststellen, welche dieser Dateien das Programm, bei dem Sie im Kommentar "Binaeres Suchverfahren" geschrieben haben, enthält. Geben Sie dazu einen möglichst einfachen Weg an!

(*e*) 6 Punkte

Sie haben einen richtigen UNIX-Filter mit Modula-2 realisiert, das ausführbare Programm heißt *doit*. Sie wollen es nun so starten, daß die Eingabe aus der Datei *file1* genommen wird, die Ausgabe in die Datei *file2* geht und die Fehlermeldungen in die Datei *err* wandern. Geben Sie dazu zwei mögliche Aufrufsequenzen an!

Aufgabe 2

20 Punkte
Schreiben Sie ein Modula-2-Programm, das in einem Text von der Standardeingabe die Häufigkeit der Buchstabenkombination **ei** feststellt.

Aufgabe 3

10 Punkte
Beschreiben Sie kurz, welche Ausgabe das folgende Programm produziert:

```
MODULE obscure;

   FROM InOut IMPORT Write, Read, WriteLn, Done;

   VAR x, y : CHAR;
       i: CARDINAL;
   CONST length = 40;


BEGIN
     i := 0;
     Read(x);
     WHILE Done DO
          INC(i);
          IF (x >= 'a') & (x <= 'z') THEN
               y:=CHR(ORD(x) - (ORD('a')-ORD('A')));
          ELSE
               y := x;
          END;
          Write(y);
          IF (i >= length) & ( (x = 40C) OR (x = 11C) ) THEN
               i := 0;
               WriteLn;
          END;
          Read(x);
     END;
END obscure.
```

Aufgabe 4

Vereinfachen Sie die folgenden beiden BOOL'schen Ausdrücke ((a), (b)):

(*a*) 4 Punkte
(a > 0) AND ((a > 2) OR TRUE)

(*b*) 4 Punkte
(a <> 0) AND NOT ((b = 2) OR NOT ((a<0) AND (b > 2)))

(*c*) 6 Punkte
Geben Sie für die Variablen a, b des folgenden Programmfragments Werte an, mit denen jeder der Zweige (THEN- / ELSE-Teile der IF-Anweisungen) durchlaufen werden. Schreiben Sie diese Werte in die Kommentarstellen!

```
VAR a,b: INTEGER;


IF (a<>0) AND NOT ( ( b > a) OR ( a < 0) )
THEN
     (*a=    , b=     *)
ELSE
     IF (b>0)
     THEN
          (*a=    , b=     *)
     ELSE
          (*a=    , b=     *)
     END;
END;
```

Aufgabe 5

10 Punkte
Eine (fiktive) UNIX-Shell erlaube folgende Kommandozeilen-Syntax:

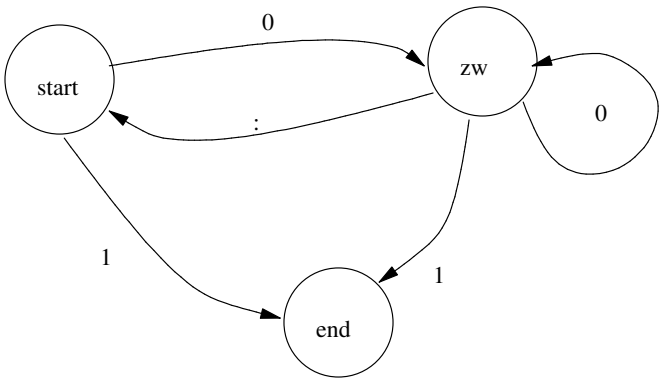
```
CmdLine   ::= [ Cmd [ ('>' File | '>>' File) ] ] '12C'
Cmd ::= Word { Word }
File      ::= Word
```

Die in einfache Hochkommas eingeschlossenen Symbole sind Terminalsymbole. Das Symbol *Word* ist zwar aus einzelnen Zeichen aufgebaut, soll hier aber nicht weiter erläutert werden. Betrachten Sie also *Word* als Terminalsymbol.

Geben Sie einen endlichen Automaten in grafischer Darstellung an (vgl. Aufgabe 6), der diese Sprache erkennt.

Aufgabe 6

Gegeben sei der folgende endliche Automat:



Die Symbole (Beschriftungen der Kanten) sind als Terminalsymbole (also: {**'1','0',':'**)zu verstehen!

(*a*) 6 Punkte
Beschreiben Sie die von diesem Automaten definierte Sprache in EBNF.

(*b*) 6 Punkte
Beschreiben Sie die von diesem Automaten definierte Sprache mit einem regulären Ausdruck.

Aufgabe 7

12 Punkte
Gegeben sei das folgende Programmfragment:

```
MODULE zustand;


TYPE
     Zustand = (z1, z2, z3, z4, z5);
VAR
     z: Zustand;
     ch: CHAR;


BEGIN
     z := z1;
     Read(ch);
     WHILE Done DO
          IF ch <> 40C THEN
               (*mit INC(z) auf den naechsten
                 Zustand in der Aufzaehlung
                 weitergehen - zyklisch:
                *)



                    INC(z);









          END;
          Read(ch);
     END;
     (* erreichten Zustand ausgeben: *)









END zustand.
```

Das Programm wandert also durch den "Zustandsraum" (sollte es zumindest tun - und zwar zyklisch).

Korrigieren Sie dieses Programm und ergänzen Sie es so, daß am Ende der Schleife der erreichte Zustand ausgegeben wird. Ihr Programm muß natürlich ein vollständiges Modula-2-Programm sein.