Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: ______ Kennwort: ______ Arbeitsplatz-Nr.: ______	**Herbst** **1998**	**46114**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

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

Einzelprüfung: **Algorithmen/Datenstrukt./Progr.-meth.**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

Bitte wenden!

Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1:

Eine Geldbörse enthält eine Menge Me von Münzen Mü (mit Wert w(Mü)).
(Die Münzen können verschiedene Werte haben.)

a) Beschreiben Sie einen Algorithmus, der zu einem Geldbetrag g eine Teilmenge T von Münzen aus Me liefert, deren Wert gerade g beträgt (falls es eine solche Teilmenge T gibt)!

b) Ist entscheidbar, ob es zu g eine Teilmenge T mit Wert g gibt? (Begründung!)

c) Formulieren Sie Ihren Algorithmus aus a) in einer Programmiersprache ihrer Wahl!

d) Beschreiben Sie die in a) bzw. c) verwendeten Datenstrukturen!

Aufgabe 2:

Geben Sie die Grammatik $G=(V,T,P,S)$ mit Variablenmenge $V=\{S,A,B\}$, terminalem Zeichenvorrat $T=\{a,b\}$, dem Startsymbol S und der Produktionenmenge $P=\{S\to AB, S\to AaBb, S\to AaaBbb, A\to aaaA, A\to\varepsilon, B\to bbbB, B\to\varepsilon\}$.

a) Welchen Typ hat die Grammatik?

b) Bestimmen Sie die von G erzeugte Sprache L(G)!

c) Ist L(G) regulär?

Aufgabe 3:

Betrachten Sie die rekursiv definierte Wortcodierungsfunktion
$\text{code}: \{a,b\}^* \to N$ mit
$\text{code}(\varepsilon)=0, \text{code}(a)=1, \text{code}(b)=2,$
$\text{code}(w\sigma)=2\cdot\text{code}(w)+\text{code}(\sigma)$ $(w\in\{a,b\}^*, \sigma\in\{a,b\})$.

a) Berechnen Sie code(abb)!

b) Zeigen Sie, daß code bijektiv ist!

c) Definieren Sie die Umkehrfunktion decode = code^{-1}, oder geben Sie einen Algorithmus für decode an!

d) Berechnen Sie decode(11)!

Thema Nr. 2

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1:

Für einen endlichen nicht-leeren Zeichenvorrat Z sei eine Binärcodierung

$$c : Z \to \{O, L\}^*$$

gegeben.

1.1 Geben Sie eine Datenstruktur (durch Typdefinitionen) an, mit deren Hilfe ein beliebiger Binärcode c im Hinblick auf die folgende Teilaufgabe 1.2 geeignet dargestellt werden kann!

1.2 Schreiben Sie eine Funktionsprozedur *fano*, mit deren Hilfe festgestellt werden kann, ob ein Binärcode c der Fano-Bedingung genügt! Die Funktionsprozedur *fano* soll dabei folgender Spezifikation genügen:

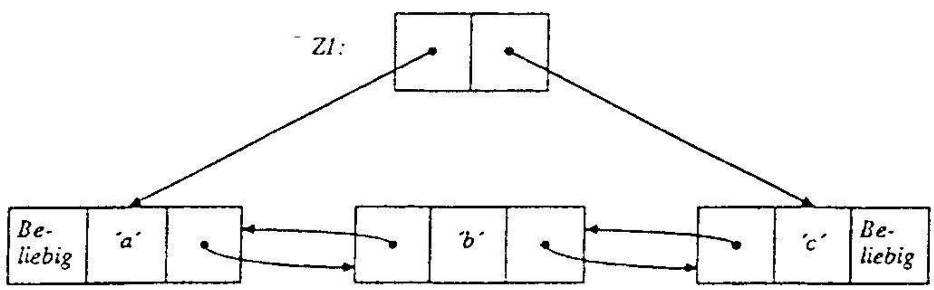
Eingabeparameter: c vom Typ *CODE*

Resultat: Das Resultat ist vom Typ *BOOLEAN* und hat den Wert *TRUE* genau dann, wenn c der Fano-Bedingung genügt.

Hinweise:
- Der Typ *CODE* muß in Ihrer Antwort zu 1.1 definiert werden.
- Ein Code erfüllt die Fano-Bedingung genau dann, wenn in ihm kein Codewort Anfang eines von ihm verschiedenen Codewortes ist.

Aufgabe 2:

Zeichenreihen über dem endlichen Alphabet $A = ('a','b',\ldots,'z')$ sollen durch vorwärts und rückwärts verkettete Listen dargestellt werden, für deren Anfang und Ende Anker (Verweise) vorgesehen sind. Die Darstellung der Zeichenreihe $z1$='abc' läßt sich also beispielsweise folgendermaßen graphisch veranschaulichen:



Fortsetzung nächste Seite!

2.1 Geben Sie eine Datenstruktur mit Hilfe geeigneter Typdefinitionen an, durch die sich die o.a. Darstellung von Zeichenreihen realisieren läßt!

2.2 Wie stellen Sie die leere Zeichenreihe ϵ dar?

2.3 Als Länge einer Zeichenreihe bezeichnet man die Anzahl der Zeichen, die sie enthält, wobei mehrfach vorkommende Zeichen entsprechend ihrer Häufigkeit gezählt werden. Die Länge der leeren Zeichenreihe ϵ ist 0.
Schreiben Sie eine Funktionsprozedur *laenge*, die die Länge einer Zeichenreihe berechnet, also folgender Spezifikation genügt:

2.3.1 Eingabeparameter: $x1$ vom Typ *ZEICHENREIHE*

Hinweis: *ZEICHENREIHE* muß als Typ in Ihrer Antwort zu 2.1 definiert sein.

2.3.2 Der Funktionswert ist vom Typ *INTEGER* und ist gleich der Länge von $x1$.

Hinweis: Für die leere Zeichenreihe ϵ soll die in Ihrer Antwort zu 2.2 angegebene Darstellung verwendet werden.

2.4 Schreiben Sie zwei Funktionsprozeduren *anfang* und *teil*, die feststellen, ob x Anfangs- bzw. Teilzeichenreihe einer Zeichenreihe y ist! Die entsprechenden Spezifikationen lauten:

2.4.1 Eingabeparameter (für beide Funktionsprozeduren): x, y vom Typ *ZEICHENREIHE*

2.4.2 Der Funktionswert ist jeweils vom Typ *BOOLEAN*. Er hat den Wert *TRUE* für die Funktion *anfang* bzw. *teil* genau dann, wenn x Anfangs- bzw. Teilzeichenreihe von y ist.

Hinweis: Es ist empfehlenswert, erst die Funktionsprozedur *anfang* zu schreiben und in der Funktionsprozedur *teil* die Funktion *anfang* zu verwenden. Die Funktionsprozedur *laenge* darf verwendet werden.

2.5 Ein Palindrom ist eine Zeichenreihe, die mit ihrer rückwärts gelesenen identisch ist. Beispiele sind

'*otto*', '*reliefpfeiler*', ϵ (leere Zeichenreihe).

Schreiben Sie eine Funktionsprozedur *palin*, die feststellt, ob eine Zeichenreihe ein Palindrom ist, also folgender Spezifikation genügt:

2.5.1 Eingabeparameter: $x1$ vom Typ *ZEICHENREIHE*

2.5.2 Die Funktion ist total, d.h. sie liefert für jede Eingabe-Zeichenreihe einen eindeutig bestimmten Wert. Der Funktionswert ist vom Typ *BOOLEAN* und ist genau dann gleich *TRUE*, wenn $x1$ ein Palindrom ist.

Hinweis: In der Funktionsprozedur *palin* darf die unter 2.3 spezifizierte Funktion *laenge* verwendet werden.

Fortsetzung nächste Seite!

Aufgabe 3:

Gegeben seien die Standard-Rechenstrukturen $\mathbb{Z}$ der ganzen Zahlen und $\mathbb{B}_2$ der Wahrheitswerte. Wie üblich seien dabei der Typ der Elemente von $\mathbb{Z}$ mit *INTEGER* und der Typ der Elemente von $\mathbb{B}_2$ mit *BOOLEAN* bezeichnet. Als Operationen sind

$+, -, *, /$ (partiell!) in $\mathbb{Z}$

$\neg, \wedge, \vee$ in $\mathbb{B}_2$

sowie die Vergleichsoperationen

$<, =$ für Elemente aus $\mathbb{Z}$

standardmäßig vorgesehen.

Durch Abstützung auf die Rechenstrukturen $\mathbb{Z}$ und $\mathbb{B}_2$ soll die im folgenden spezifizierte Teil-Rechenstruktur $\mathbb{Q}_1$ der rationalen Zahlen $\mathbb{Q}$ vollständig dargestellt werden.

3.1 Geben Sie eine Datenstruktur an, mit deren Hilfe sich jede Zahl aus $\mathbb{Q}$ im Hinblick auf die folgenden Teilaufgaben geeignet darstellen läßt! Geben Sie außerdem an, welchen Einschränkungen die Objekte der von Ihnen vorgeschlagenen Datenstruktur genügen sollen, damit sie in $\mathbb{Q}$ zulässig sind!

3.2 Schreiben Sie Prozeduren für die Multiplikation (*mult*) und die Division (*div*) in $\mathbb{Q}$. Diese Prozeduren sollen den folgenden Spezifikationen genügen:

3.2.1 Eingabeparameter: a, b: *RATIONAL*
Dabei dürfen Sie unterstellen, daß die von Ihnen in Ihrer Antwort zu 3.1 angegebenen Einschränkungen für in $\mathbb{Q}$ zulässige Objekte von a und b erfüllt sind! Bei *div* dürfen Sie außerdem unterstellen, daß b nicht die Null darstellt!

3.2.2 Resultatparameter: `VAR` c: *RATIONAL*
Der Wert von c muß nach Ausführung von *mult* gleich $a * b$ und nach Ausführung von *div* gleich a/b sein. Dabei muß dieser Wert jeweils den Zulässigkeitseinschränkungen, die Sie in Ihrer Antwort zu 3.1 formuliert haben, genügen!

3.3 Schreiben Sie eine Funktionsprozedur *gr*, die feststellt, ob eine Zahl a aus $\mathbb{Q}$ größer ist als eine Zahl b aus $\mathbb{Q}$! Dabei soll *gr* folgender Spezifikation genügen:

3.3.1 Eingabeparameter: a, b: *RATIONAL*
Sie dürfen wieder unterstellen, daß a und b den Zulässigkeitseinschränkungen gemäß 3.1 genügen!

3.3.2 Das Resultat ist vom Typ *BOOLEAN* und hat den Wert *TRUE* genau dann, wenn $a > b$ ist.