FREISTAAT THÜRINGEN
Kultusministerium

ABITURPRÜFUNG 2007

LEISTUNGSFACH

INFORMATIK

(HAUPTTERMIN)

Arbeitszeit: 270 Minuten

Hilfsmittel: Wörterbuch zur deutschen Rechtschreibung
Taschenrechner (nicht programmierbar, nicht grafikfähig)
(Schüler, die einen CAS-Taschencomputer im Unterricht benutzen, dürfen diesen verwenden.)
Tafelwerk
PC mit Prolog-System und
Oberon-, Pascal- oder Java-System
Zufallszahlengenerator

Lösen Sie die Aufgaben 1 und 2 und wählen Sie von den Aufgaben 3.1 und 3.2 eine Aufgabe zur Bearbeitung aus.

Rechts unten neben jeder Teilaufgabe steht die für diese Teilaufgabe maximal erreichbare Anzahl von Bewertungseinheiten (BE).

Der Prüfungsteilnehmer sichert bei der praktischen Arbeit am PC mindestens alle 10 Minuten die von ihm erarbeiteten Quelltexte. Er hat die von ihm erarbeiteten Programme und Module im Quelltext zu kommentieren.
Die Quelltexte sind zusammen mit der Abiturarbeit abzugeben.

ÖFFNUNG AM 09. MAI 2007

Aufgabe 1

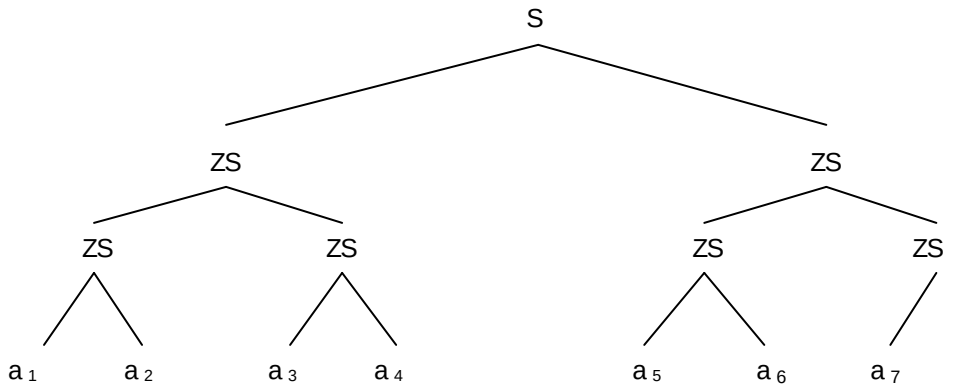
1.1 Auf dem Bildschirm eines Computers wird 72,345 ausgegeben. Erläutern Sie am Beispiel dieser Ausgabe, dass eine Information aus einem syntaktischen und einem semantischen Teil besteht. 2 BE

1.2 Setzen Sie sich mit der These Joseph Weizenbaums auseinander:
„Der Mensch ist keine Maschine. Ich werde behaupten, dass der Mensch zwar zweifelsohne Informationen verarbeitet, dass er dies jedoch nicht unbedingt in derselben Weise tun muss wie Computer. Computer und Menschen sind nicht verschiedene Arten derselben Gattung.“ 5 BE

1.3 Gegeben sind eine positive ganze Zahl n und die Zahlen a_1 bis a_n.

1.3.1 Geben Sie einen Algorithmus an, der die Summe S der Zahlen a_1 bis a_n berechnet, indem er die Zahlen nacheinander addiert. Stellen Sie den Algorithmus in Form eines Struktogramms dar.

1.3.2 Die Summe S der Zahlen a_1 bis a_n kann auch so berechnet werden, wie es in der folgenden Abbildung für sieben Zahlen grafisch dargestellt ist:



In der Abbildung sind ZS Zwischensummen.

Geben Sie einen Algorithmus an, der die Summe S so berechnet, dass mehrere Zwischensummen zur gleichen Zeit gebildet werden.
Vergleichen Sie die Zeitkomplexität des Algorithmus aus Teilaufgabe 1.3.1 mit der des Algorithmus aus Teilaufgabe 1.3.2.

5 BE

1.4 Erläutern Sie, ob der Algorithmus aus Teilaufgabe 1.3.2 in einer Ihnen bekannten Programmiersprache implementiert werden kann. 3 BE

Aufgabe 2

Der folgende Text enthält Fakten über Gestalten aus der griechischen Mythologie:
Europa, Aithra, Pasiphae, Phaidra und Ariadne sind weiblich. Zeus, Minos, Helios, Aigeus, Minotauros, Sarpedon, Androgeos, Tauros, Rhadamanthys und Theseus sind männlich. Theseus ist das Kind von Aigeus. Sarpedon, Rhadamanthys und Minos sind die Kinder von Zeus. Androgeos, Minotauros, Phaidra und Ariadne sind die Kinder von Pasiphae. Minos, Rhadamanthys und Sarpedon sind die Kinder von Europa. Pasiphae ist das Kind von Helios. Ariadne, Phaidra und Androgeos sind die Kinder von Minos. Minotauros ist das Kind von Tauros. Theseus ist das Kind von Aithra.

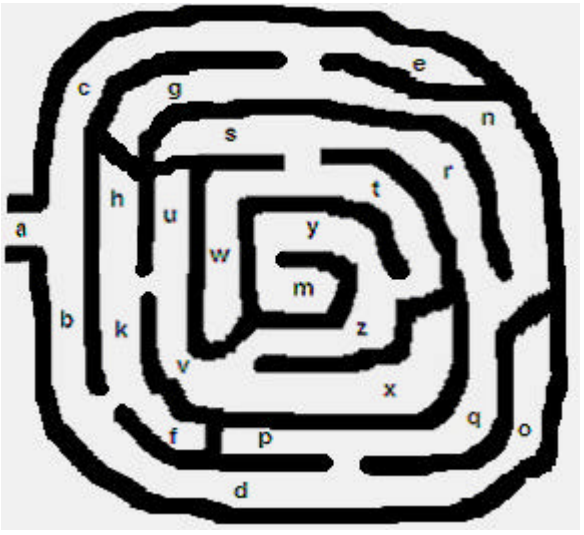
Übertragen Sie die Fakten aus dem Text in ein Prolog-Programm. Sie dürfen im Programm für die Namen geeignete Abkürzungen verwenden.

Zwei weibliche Gestalten hatten denselben Halbbruder. Der Halbbruder war ein Ungeheuer und hielt sich in einem Labyrinth verborgen.
Erweitern Sie Ihr Programm so, dass das Prolog-System mit Hilfe der gegebenen Fakten die Antworten auf die Fragen findet:

- Wie heißen die weiblichen Gestalten, deren Halbbruder ein Ungeheuer war?
- Wie heißt der Vater des Ungeheuers?

Erläutern Sie, wie das Prolog-System die Antworten auf diese Fragen findet.

Theseus fand im Labyrinth den Weg zum Ungeheuer und besiegte es. In der Abbildung ist ein solches Labyrinth dargestellt. Jeder Gang des Labyrinths ist mit einem Buchstaben gekennzeichnet.



Entwerfen Sie in Prolog eine Suche, die im abgebildeten Labyrinth alle Wege vom Eingang a bis zum Gang m findet. Beachten Sie dabei, dass kein Gang mehrmals durchlaufen werden darf. Für jeden gefundenen Weg sollen, beginnend mit dem Eingang a, die Gänge in der richtigen Reihenfolge ausgegeben werden.

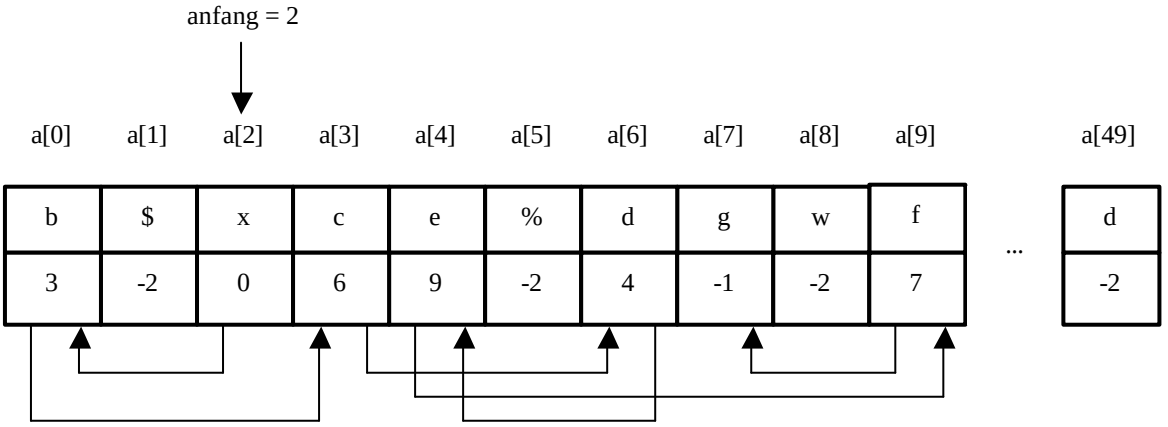
Implementieren Sie die Wegsuche in Ihrem Prolog-Programm.

15 BE

Aufgabe 3.1

Vor 35 Jahren legte David Lorge Parnas den Grundstein für die modulare Programmierung und das Geheimnisprinzip (information hiding). Erklären Sie die Begriffe modulares Programmieren und Geheimnisprinzip.

Eine einfach verkettete Liste kann mit Hilfe einer Reihung (Array) realisiert werden. In der folgenden Abbildung ist eine solche Liste grafisch dargestellt:



Jedes Element der Reihung kann ein Knoten der Liste sein. Jeder Knoten besteht aus zwei Komponenten. Die erste Komponente enthält ein Zeichen. Die zweite Komponente enthält eine Zahl. Diese Zahl ist -1 bzw. -2 oder die Nummer des folgenden Knotens der Liste. Die Zahl -1 kennzeichnet das Ende der Liste. Die Zahl -2 kennzeichnet einen freien Knoten.
Ist die Liste leer, hat die Variable anfang den Wert -1. Anderenfalls hat die Variable anfang den Wert der Nummer des ersten Knotens der Liste.

Entwerfen und implementieren Sie ein Modul Liste, das genau eine Liste mit den folgenden Operationen realisiert:

Operation	Beschreibung
erzeugen	Die Operation erzeugt eine leere Liste.
leer	Die Operation prüft, ob die Liste leer ist.
einfuegen(e)	Die Operation fügt den Wert e in einen freien Knoten der Liste ein.
suchen(e)	Die Operation prüft, ob der Wert e in einem Knoten der Liste enthalten ist.
loeschen(e)	Falls der Wert e in einem bzw. mehreren Knoten der Liste enthalten ist, entfernt die Operation den bzw. die Knoten aus der Liste und gibt den bzw. die entfernten Knoten frei.

Beachten Sie die Festlegungen:
- Zur Realisierung der Liste ist die vorgegebene Beschreibung einer einfach verketteten Liste zu verwenden.
- Die Anzahl der Elemente der Reihung ist 50.
- Das Geheimnisprinzip ist im Modul Liste zu realisieren.

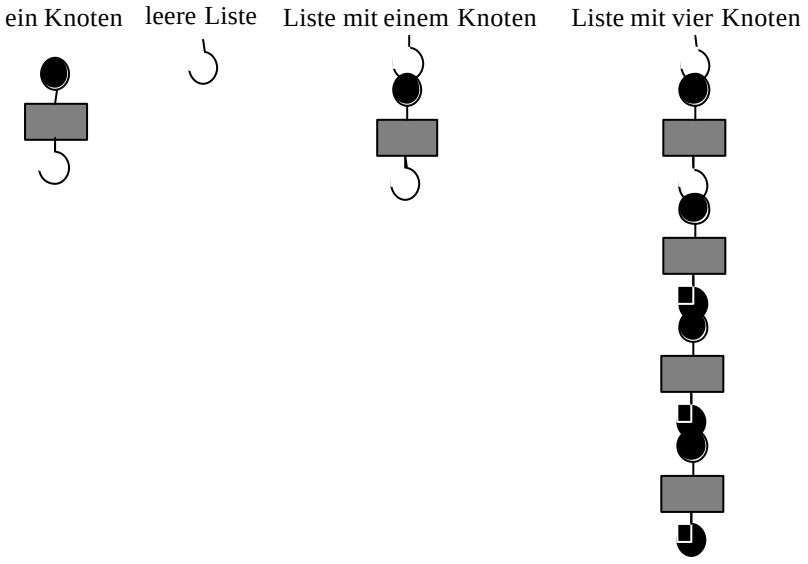
Implementieren Sie ein Programm, das Folgendes leistet:
– Erzeugen einer leeren Liste.
– Einfügen von Zeichen in die Liste.
– Prüfen, ob gesuchte Zeichen in der Liste enthalten sind.
– Löschen von Zeichen aus der Liste.
– Prüfen, ob die Liste leer ist.

Das Modul Liste ist in Ihr Programm zu importieren und zu verwenden.

Testen Sie Ihr Programm durch folgende Handlungen:
– Einfügen der Zeichen i, n, f, o, r, m, a, t, i, k in die Liste.
– Prüfen, ob die Zeichen i, o und u in der Liste enthalten sind.
– Löschen der Zeichen i und t aus der Liste.
– Prüfen, ob die Zeichen i und t in der Liste enthalten sind.
– Löschen aller Zeichen aus der Liste.
– Prüfen, ob die Liste leer ist.

Dokumentieren Sie den Test.

In der folgenden Abbildung ist ein Modell einer einfach verketteten Liste dargestellt:



Erläutern Sie an diesem Modell die Aussage von Karl Steinbuch:
„Modelle unterscheiden sich in irgendwelchen Merkmalen von ihren Originalen – sonst wären sie nicht deren Modelle, aber sie haben auch manches mit ihnen gemeinsam – sonst wären sie nicht deren Modelle.“

30 BE

Aufgabe 3.2

Auf einem 8×8 Schachbrett sollen acht Türme so aufgestellt werden, dass jede Zeile und jede Spalte des Schachbretts genau einen Turm enthält.
Beschreiben Sie, wie mit Backtracking die 40320 verschiedenen Aufstellungen der Türme gefunden werden können.

Sudoku ist ein japanisches Zahlenrätsel, das aus einem großen Quadrat aus neun Zeilen und neun Spalten besteht. Weiterhin ist das große Quadrat in neun kleine Quadrate eingeteilt. Jedes kleine Quadrat besteht aus drei Zeilen und drei Spalten. Einige Felder der Quadrate enthalten Zahlen, wie beispielsweise in der Abbildung dargestellt:

				6		2	9	5
	2	4			8			1
	6		1			8		4
		9		4	3		8	
4			8		9			2
	8		7	5		4		
7		5			1		2	
3			5			1	7	
6	1	8		7				

Beim Sudoku müssen die leeren Felder des großen Quadrates so ausgefüllt werden, dass in jeder Zeile und jeder Spalte jede Zahl von 1 bis 9 steht. Außerdem muss auch jedes kleine Quadrat jede Zahl von 1 bis 9 enthalten.
Entwerfen und implementieren Sie ein Programm, das für das in der Abbildung dargestellte Sudoku genau eine Lösung sucht. Wird eine Lösung gefunden, dann ist diese auszugeben.

30 BE