

ABITURPRÜFUNG 2003

LEISTUNGSFACH

INFORMATIK (HAUPTTERMIN)

Arbeitszeit: 270 Minuten

Hilfsmittel: Formeln und Tabellen für die Sekundarstufen I und II.
Berlin: Paetec, Ges. für Bildung und Technik mbH;
PC mit Prolog-System und
Oberon- oder Turbo Pascal-System;
Zufallszahlengenerator; Realisation des ADT Liste;
Taschenrechner

Der Prüfungsteilnehmer löst die Aufgaben 1 und 2 und wählt 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 Abiturnarbeit abzugeben.

ÖFFNUNG AM 07. MAI 2003

Aufgabe 1

- 1.1 Erläutern Sie ein Verfahren zur Datenkompression!

3 BE

- 1.2 Entwerfen Sie einen Algorithmus, der die Suche eines Musters in einem Text ausführt! Der Algorithmus soll die Position des Musters im Text ermitteln.

Beispiel:

Muster: Mensch

Text: Die Geschichte des Menschen und
die seiner Maschinen sind
untrennbar ineinander verwoben.

Ergebnis: Der Buchstabe M im Wort Menschen ist das
20. Zeichen im Text. Der Algorithmus ermittelt
als Position den Wert 20.

4 BE

- 1.3 Computernetze werden in WAN und LAN eingeteilt.

Erläutern Sie am Beispiel Internet, was man unter einem WAN versteht!

Erläutern Sie am Beispiel des Computernetzes einer Schule, was man unter einem LAN versteht!

4 BE

- 1.4 Erläutern Sie, wie eine E-Mail von einem Sender-PC zu einem Empfänger-PC über das Internet übertragen wird!

4 BE

Aufgabe 2

Ein gemeiner Bruch ist ein geordnetes Paar zweier ganzer Zahlen, wobei die zweite Zahl nicht 0 sein darf. Beispiele für gemeine Brüche sind $(3; 4)$, $(-7; 4)$, $(-7; +4)$ und $(-8; -9876452)$.

Das geordnete Paar $(47; 0)$ ist kein gemeiner Bruch.

- a) Geben Sie eine Definition der Syntax einer ganzen Zahl in dem erweiterten Backus-Naur-Formalismus (EBNF) an!

Geben Sie eine Definition der Syntax eines gemeinen Bruches im EBNF an!

Überführen Sie die beiden Definitionen aus dem EBNF in Syntaxdiagramme!

Erläutern Sie am Beispiel "gemeiner Bruch", was man unter Syntax und Semantik versteht!

7 BE

- b) Entwerfen und implementieren Sie ein Prolog-Programm, das erkennt, ob ein bei der Anfrage angegebener gemeiner Bruch syntaktisch korrekt ist!

Grundlage des Programms ist die Definition der Syntax eines gemeinen Bruches von Teilaufgabe a).

In der Anfrage ist der gemeine Bruch als Liste anzugeben.

Ein Beispiel ist der gemeine Bruch $(-30124; 2873551)$:

?- bruch(['-',3,0,1,2,4,',';2,8,7,3,5,5,1]).

8 BE

Aufgabe 3.1

3.1.1 Einfach verkettete Listen können mit Hilfe des Zeigertyps realisiert werden.

a) Erläutern Sie den Unterschied zwischen Zeiger- und Bezugsvariable!

2 BE

b) Gegeben sind die folgenden Typ- und Variablendeklarationen:

Oberon	Turbo Pascal
TYPE tzeiger = POINTER TO tknoten; tknoten = RECORD d: INTEGER; z: tzeiger END; VAR p, q: tzeiger;	TYPE tzeiger = ^tknoten; tknoten = RECORD d: integer; z: tzeiger END; VAR p, q: tzeiger;

Erläutern Sie, was die folgenden Anweisungen leisten!

Gehen Sie in Ihrer Antwort entweder auf Oberon oder auf Turbo Pascal ein!

Oberon	Turbo Pascal
NEW(p); p^.d := 12; p^.z := NIL; q := p; IF p = q THEN Out.String("ja") ELSE Out.String("nein") END	NEW(p); p^.d := 12; p^.z := NIL; q := p; IF p = q THEN write('ja') ELSE write('nein')

6 BE

c) Definieren Sie den Begriff „einfach verkettete Liste“!

- 3.1.2 An einem Versteckspiel nehmen n Kinder teil. Die Kinder erhalten die Nummern von 1 bis n . Zuerst wird das Kind, das die anderen zu suchen hat, festgelegt. Dies geschieht durch mehrmaliges Abzählen mit Hilfe eines Reims. Der Reim besteht aus k Wörtern. Es gilt $k < n$. Vor dem Abzählen stellen sich die Kinder in der Reihenfolge von 1 bis n im Kreis auf. Links von Kind 1 steht Kind 2. Links von Kind 2 steht Kind 3 usw. Links von Kind n steht Kind 1. Das Abzählen erfolgt stets links herum.

Das erste Abzählen beginnt mit Kind 1, wird mit den Kindern von 2 bis $k-1$ fortgeführt und endet mit Kind k . Kind k verlässt den Kreis und versteckt sich. Anschließend wird zum zweiten Mal mit Hilfe des Reims abgezählt. Das Abzählen beginnt jetzt mit Kind $k+1$. Das Kind, bei dem das Abzählen endet, verlässt den Kreis und versteckt sich. Das Abzählen wird so oft wiederholt, bis nur noch ein Kind im Kreis steht. Dieses Kind hat die anderen Kinder zu suchen.

Beispiel ($n = 7, k = 4$):

Die Kinder verlassen den Kreis in der Reihenfolge 4, 1, 6, 5, 7 und 3 und verstecken sich. Kind 2 hat die anderen Kinder zu suchen.

Entwerfen und implementieren Sie ein Programm in Oberon oder Turbo Pascal, das die beschriebenen Vorgänge zum Abzählen simuliert! Der Kreis ist mit Hilfe einer einfach verketteten Ringliste zu realisieren. Bei einer Ringliste gibt es einen Verweis vom letzten auf das erste Listenelement. Die Ringliste ist mit Hilfe des Zeigertyps zu realisieren.

Aufgabe 3.2

3.2.1 In Programmiersprachen wie Oberon oder Turbo Pascal ist der Begriff des Datentyps von grundlegender Bedeutung.

a) Erläutern Sie, was man unter einem Datentyp versteht!

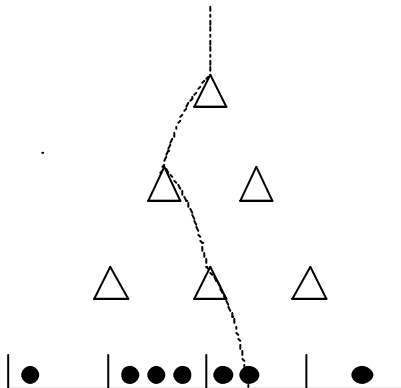
2 BE

b) Wählen Sie von den einfachen Datentypen INTEGER, REAL, CHAR und BOOLEAN zwei aus und geben Sie für diese Datentypen wesentliche Operationen und Relationen sowie das Prinzip der internen Realisation an!

4 BE

3.2.2 Ein Galtonbrett besteht aus mehreren Reihen Hindernissen, die versetzt untereinander angeordnet sind. In jeder Reihe kommt ein Hindernis dazu. Eine Kugel fällt von oben nach unten durch das Galtonbrett. Bei jedem Hindernis setzt die Kugel mit einer Wahrscheinlichkeit von jeweils 0,5 ihren Weg links bzw. rechts fort. Unter den Zwischenräumen der letzten Reihe befinden sich Behälter. Die Kugel wird in einem der Behälter aufgefangen.

Die folgende Abbildung zeigt den Weg einer Kugel durch ein Galtonbrett mit drei Reihen.



- a) Entwerfen Sie eine Funktion, die eine Zufallszahl aus der Menge $\{0, 1\}$ ermittelt! Beide Zahlen sollen mit gleicher Wahrscheinlichkeit gezogen werden.

Implementieren Sie die Funktion in Oberon oder Turbo Pascal und testen Sie die Funktion!

Dokumentieren Sie die Tests!

4 BE

- b) Entwerfen Sie ein Programm, das die Arbeitsweise eines Galtonbretts simuliert!
 Durch das Galtonbrett sollen n Kugeln fallen.
 Das Programm liest die Anzahl r an Reihen und die Anzahl n an Kugeln ein und gibt aus, wie viele Kugeln in den verschiedenen Behältern aufgefangen wurden.

Es gilt $1 \leq r \leq 10$ und $1 \leq n \leq 20000$.

Implementieren Sie das Programm in Oberon oder Turbo Pascal!

Dokumentieren Sie die Ergebnisse der Programmabarbeitung für mehrere Werte von r und n !

10 BE

- 3.2.3 Entwerfen Sie ein Programm, das Binomialkoeffizienten $\text{bino}(n, k)$ mit Hilfe der folgenden Erläuterungen berechnet!
 n und k sind ganze Zahlen mit $n \geq k \geq 0$.

Sind $k = 0$ oder $n = k$, so gilt $\text{bino}(n, k) = 1$.

Ansonsten gilt $\text{bino}(n, k) = \text{bino}(n - 1, k - 1) + \text{bino}(n - 1, k)$.

Implementieren Sie das Programm in Oberon oder Turbo Pascal! Testen Sie das Programm für mehrere Werte von n und k ! Dokumentieren Sie die Tests!

Erläutern Sie an dem implementierten Programm, was man unter rekursiven Unterprogrammen versteht!

10 BE
