



FREISTAAT THÜRINGEN

Kultusministerium



ABITURPRÜFUNG 2002

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;
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 Abiturarbeit abzugeben.

ÖFFNUNG AM 24. APRIL 2002

Aufgabe 1

- a) Definieren Sie die Begriffe "binärer Baum" und "binärer Suchbaum"!

4 BE

- b) Gegeben ist ein leerer binärer Suchbaum.
Die folgenden 14 Wörter werden nach einander in den Suchbaum eingefügt:

OF, PROCEDURE, THEN, END, REPEAT, FOR,
UNTIL, DO, CASE, BEGIN, TO, WHILE, ELSE, IF

Geben Sie den binären Suchbaum nach dem Einfügen der Wörter an!

3 BE

- c) Erläutern Sie, wie die folgenden drei Operationen in einem binären Suchbaum auszuführen sind:

- Einfügen eines Knotens,
- Suchen eines Elementes und
- Löschen eines Knotens.

4 BE

- d) Entwerfen Sie einen Algorithmus, der Folgendes leistet:

Die Elemente eines binären Suchbaumes werden in lexikographischer Reihenfolge ausgegeben.

4 BE

Aufgabe 2

Eine Pension verfügt über neun Einzelzimmer. Die Zimmer haben die Nummern von 1 bis 9. An der Rezeption wird eine Tabelle geführt, in der zu jedem Zimmer der Name des Gastes vermerkt wird. Ist ein Zimmer nicht belegt, steht an Stelle des Namens der Vermerk „frei“. Keine zwei Gäste besitzen den gleichen Namen.

Das folgende Beispiel gibt die Tabelle zu einem bestimmten Zeitpunkt an:

Zimmer	Gast
1	Meier
2	frei
3	frei
4	Schulze
5	Hinze
6	frei
7	Kunze
8	frei
9	frei

Entwerfen und implementieren Sie ein Prolog-Programm, das Folgendes leistet:

- Verwalten der Tabelle: Wird ein freies Zimmer bezogen, so wird der Vermerk „frei“ durch den Namen des Gastes ersetzt. Zieht ein Gast aus einem Zimmer aus, so wird der Name des Gastes durch den Vermerk „frei“ ersetzt. Zu Beginn sind alle Zimmer frei.
- Beantworten von Fragen zur Tabelle: Welche Zimmernummer hat ein gesuchter Gast? Wie viele Zimmer sind frei? Wie lauten die Nummern der freien Zimmer?

Geben Sie zu Ihrem Prolog-Programm die notwendigen Anfragen zum Verwalten der Tabelle und Beantworten der Fragen an!

Aufgabe 3.1

In vielen Computerprogrammen werden Kalenderdaten genutzt.

Wichtige Festlegungen für Kalenderdaten sind: Die Monate Januar, März, Mai, Juli, August, Oktober und Dezember haben 31 Tage. Die Monate April, Juni, September und November haben 30 Tage. Der Februar hat 29 Tage, falls er in einem Schaltjahr liegt. Ansonsten hat er 28 Tage. Ein Jahr ist ein Schaltjahr, wenn die Jahreszahl ohne Rest durch 400 teilbar ist oder wenn die Jahreszahl ohne Rest durch 4 und nicht durch 100 teilbar ist. Ein Schaltjahr hat 366 Tage. Die anderen Jahre haben 365 Tage.

Im Weiteren wird nur der Zeitraum vom 1. Januar 1701 bis zum 31. Dezember 2100 betrachtet. Die Tage werden fortlaufend nummeriert. Die Nummerierung beginnt mit dem 1. Januar 1701. Diesem Datum wird die Tagesnummer 1 zugeordnet. Dem 2. Januar 1701 wird die Tagesnummer 2 zugeordnet usw. Ein weiteres Beispiel ist der 23. Juni 2002 (90. Geburtstag von Alan Turing). Diesem Datum wird die Tagesnummer 110112 zugeordnet.

- a) Geben Sie eine Datenstruktur an, mit der ein Kalenderdatum gespeichert werden kann! Begründen Sie Ihre Antwort! Erläutern Sie, wie die Datenstruktur intern realisiert wird!

4 BE

- b) Entwerfen Sie ein Modul, das die folgenden Operationen realisiert:

- Ermitteln, ob ein gegebenes Jahr ein Schaltjahr ist,
- Ermitteln der Monatslänge für einen gegebenen Monat in einem gegebenen Jahr,
- Ermitteln der Jahreslänge für ein gegebenes Jahr,
- Ermitteln der Tagesnummer für ein gegebenes Datum und
- Ermitteln des Datums für eine gegebene Tagesnummer.

Implementieren Sie das Modul in Oberon oder Turbo Pascal!

14 BE

- c) Vom 15. März 2002 bis zum 24. April 2002 sind es 40 Tage. Der 15. März 2002 war ein Freitag. Also ist der 24. April 2002 ein Mittwoch.

Entwerfen Sie ein Programm, das das Modul von Teilaufgabe b) importiert und bei der Lösung der folgenden Aufgaben nutzt:

- Ermitteln der Anzahl Tage von einem Datum zu einem anderen Datum und
- Ermitteln des Wochentages für ein bestimmtes Datum.

Implementieren Sie das Programm in Oberon oder Turbo Pascal!

8 BE

- d) Erläutern Sie Vorteile des modularen Programmierens! Nehmen Sie in Ihrer Erläuterung Bezug auf das Modul von Teilaufgabe b) und das Programm von Teilaufgabe c) !

4 BE

Aufgabe 3.2

3.2.1 Reihungen sind Datenstrukturen, die beim Problemlösen mit imperativen Programmiersprachen vielfältig eingesetzt werden.

- a) Nennen und erläutern Sie Operationen und Relationen, die für Reihungen vorgesehen sind!

4 BE

- b) Erläutern Sie den Algorithmus "lineares Suchen"!

Geben Sie die Zeitkomplexität des Algorithmus "lineares Suchen" für den besten, den mittleren und den schlechtesten Fall an! Begründen Sie Ihre Antworten!

3 BE

- c) Erläutern Sie den Algorithmus "binäres Suchen"!

Geben Sie die Zeitkomplexität des Algorithmus "binäres Suchen" für den besten, den mittleren und den schlechtesten Fall an! Begründen Sie Ihre Antworten!

4 BE

- d) Gegeben ist ein Algorithmus, der das Minimum von 100 Zahlen ermittelt, die als eindimensionale Reihung vorliegen.

Gesucht ist ein Algorithmus, der das Minimum von 100^3 Zahlen ermittelt, die als dreidimensionale Reihung vorliegen. Das Modell der Reihung ist ein Würfel, der die Seitenlänge 100 besitzt und aus 100^3 Einheitswürfeln zusammengesetzt ist. In jedem Einheitswürfel steht eine Zahl.

Entwerfen Sie einen Algorithmus, der das gesuchte Minimum der 100^3 Zahlen ermittelt und dabei den gegebenen Algorithmus verwendet!

4 BE

3.2.2 A und B experimentieren gern mit Zahlen. A zeichnet die folgende Tabelle mit 8 Zeilen und 10 Spalten auf. Auf jeden Tabellenplatz schreibt er eine positive ganze Zahl.

23	34	12	45	21	37	89	100	29	98
43	78	18	47	74	65	29	78	82	76
6	81	154	13	37	61	70	87	38	77
14	62	44	46	57	13	75	76	32	29
30	121	69	64	26	25	91	99	58	71
65	99	16	78	11	55	93	87	77	36
87	40	38	117	49	68	39	78	57	48
9	33	145	39	50	35	63	97	105	85

B betrachtet die Zahlen und stellt fest, dass die Tabelle eine besondere Eigenschaft besitzt. Die Tabelle enthält nämlich eine Zahl, die die kleinste in ihrer Spalte und die größte in ihrer Zeile ist. "S-Zahl" nennt B eine solche Zahl.

B zeichnet die folgende Tabelle mit 7 Zeilen und 8 Spalten auf. Auch B schreibt auf jeden Tabellenplatz eine positive ganze Zahl.

45	21	85	31	50	121	18	62
23	67	55	25	17	40	26	44
36	102	111	71	39	62	13	81
98	27	78	57	93	27	26	28
34	13	47	58	56	83	40	43
28	99	10	85	83	99	89	34
109	78	95	100	35	73	37	78

A betrachtet die Zahlen und stellt fest, dass diese Tabelle die besondere Eigenschaft nicht besitzt. In der Tabelle gibt es keine S-Zahl.

Entwerfen und implementieren Sie ein Oberon- oder Turbo Pascal-Programm, das für eine Tabelle entscheidet, ob sie die besondere Eigenschaft besitzt! Besitzt die Tabelle die besondere Eigenschaft, so ist die S-Zahl auszugeben. Ansonsten ist „nein“ auszugeben. Testen Sie das Programm und dokumentieren Sie die Tests!