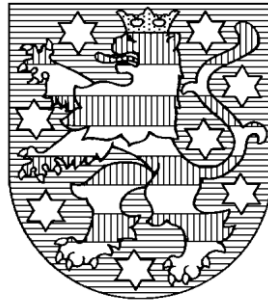


Thüringer Kultusministerium



Abiturprüfung 1998

Informatik

als Leistungsfach
(Haupttermin)

Arbeitszeit: 240 Minuten

Einlesezeit: 30 Minuten

Hilfsmittel: Formeln und Tabellen für die Sekundarstufen I und II:
Gymnasien, Gesamtschulen, Berufsschulen.
paetec Ges. für Bildung und Technik Berlin;
PC mit Prolog-System und
Turbo Pascal- oder Oberon-System;
Taschenrechner

Der Prüfungsteilnehmer löst **alle** Aufgaben.

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 Programme.

Der Prüfungsteilnehmer hat die von ihm erarbeiteten Quelltexte abzugeben.

1 Datenstrukturen

1.1 Definieren Sie den Begriff "einfach verkettete Liste" rekursiv!

2 BE

1.2 Gegeben ist eine einfach verkettete Liste. Die Listenelemente liegen in sortierter Reihenfolge vor.
Beschreiben Sie, wie in diese Liste ein neues Listenelement an der richtigen Stelle eingefügt werden kann!

4 BE

1.3 Definieren Sie den Begriff "einfach verkettete Ringliste"!

2 BE

1.4 Erläutern Sie, wie eindimensionale und zweidimensionale Reihungen (Arrays) intern in Turbo Pascal oder Oberon realisiert werden!

3 BE

1.5 Gegeben ist das folgende Zahlenschema:

```

a11
a21  a22
a31  a32  a33
a41  a42  a43  a44
a51  a52  a53  a54  a55
.....
an1  an2  an3  an4  an5  ...  ann

```

Erläutern Sie, wie ein derartiges Zahlenschema speichereffizient in einem Turbo Pascal- oder Oberon-Programm verwaltet werden kann!

4 BE

2 Verwaltung eines Warenlagers

Ein Warenlager besteht aus 2000 Lagerplätzen, die in einer Reihe angeordnet sind. Die Lagerplätze sind in dieser Reihe fortlaufend von 1 bis 2000 nummeriert.

In dem Warenlager können Kartons der Größen A, B und C aufbewahrt werden.

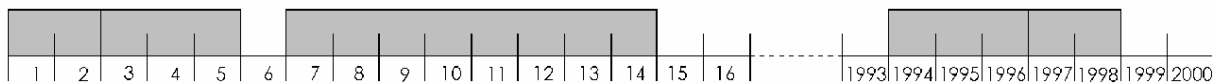
Ein Karton der Größe A belegt zwei Lagerplätze.

Ein Karton der Größe B belegt drei Lagerplätze.

Ein Karton der Größe C belegt acht Lagerplätze.

Jeder Karton besitzt eine ganzzahlige Karton-Nummer, die ihn eindeutig identifiziert. Die Karton-Nummern stammen aus dem Intervall von 1 bis 10000. Es kann vorausgesetzt werden, daß keine zwei Kartons die gleiche Karton-Nummer besitzen.

Die folgende Abbildung zeigt das Warenlager, das fünf Kartons enthält:



2.1 Entwerfen Sie ein Modul, das die folgenden Operationen zur Verwaltung des Warenlagers realisiert:

- Einlagern eines Kartons,
- Auslagern eines Kartons,
- Ermitteln, ob sich ein Karton mit einer bestimmten Karton-Nummer im Warenlager befindet!

Implementieren Sie das Modul in **Turbo Pascal** oder **Oberon**!

Beachten Sie die folgenden Festlegungen:

- Beim Einlagern eines Kartons mit einer bestimmten Karton-Nummer ist zu registrieren, auf welchen Lagerplätzen der Karton abgelegt wird.
- Ein Karton kann eingelagert werden, wenn dafür genügend viele freie Lagerplätze vorhanden sind. Im Bedarfsfall muß im Warenlager Platz geschaffen werden. Dies geschieht durch Umordnen bereits eingelagerter Kartons. Sollte der Karton dennoch nicht einzulagern sein, ist eine entsprechende Ausschrift vom PC auszugeben.
- Ist ein Karton, der ausgelagert werden soll, nicht im Warenlager enthalten, ist eine entsprechende Ausschrift vom PC auszugeben!

2.2 Entwerfen Sie ein Simulationsprogramm, das die folgende Handlungsfolge realisiert:

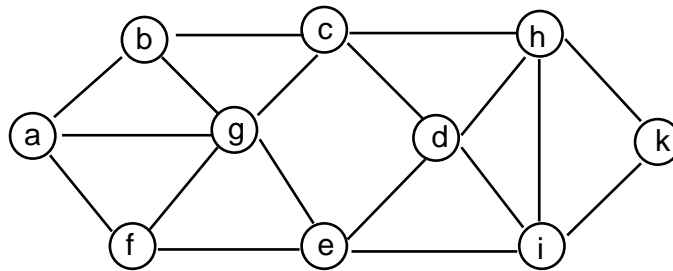
1. Handlung: Einlagern von 200 Kartons der Größe B mit den Karton-Nummern von 101 bis 300.
2. Handlung: Einlagern von 300 Kartons der Größe A mit den Karton-Nummern von 501 bis 800.
3. Handlung: Einlagern von 80 Kartons der Größe C mit den Karton-Nummern von 1001 bis 1080,
4. Handlung: Auslagern der 150 Kartons mit den Karton-Nummern von 601 bis 750.
5. Handlung: Einlagern von 50 Kartons der Größe C mit den Karton-Nummern von 2001 bis 2050.

Implementieren Sie das Programm in **Turbo Pascal** oder **Oberon**! Verwenden Sie dazu das Modul, das von Ihnen in Teilaufgabe 2.1 implementiert wurde! Gehen Sie davon aus, daß das Warenlager zu Beginn leer ist.

10 BE

3 Ungerichteter Graph

Gegeben ist der folgende ungerichtete Graph G :



Der Graph G besteht aus Knoten und Kanten.

Die Knoten sind a, b, c, d, e, f, g, h, i und k.

Jede Kante verbindet zwei Knoten des Graphen G miteinander.

Entwerfen und implementieren Sie ein **Prolog-Programm**, das im Graphen G alle Wege von einem Start- zu einem Zielknoten sucht! Formulieren Sie dazu eine Abfrage in Prolog!

Beachten Sie die folgenden Festlegungen:

- Jeder Knoten des Graphen G kann Startknoten sein.
- Jeder Knoten des Graphen G kann Zielknoten sein.
- Jeder Knoten des Graphen G darf höchstens einmal durchlaufen werden.
- Für jeden gefundenen Weg sind in der richtigen Reihenfolge der Startknoten, die Knoten, die auf dem Weg liegen, und der Zielknoten auszugeben.
- Kommentieren Sie Ihr Prolog-Programm im Quelltext!

15 BE
