

FREISTAAT THÜRINGEN

Kultusministerium



# ABITURPRÜFUNG 2001

## 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  
Turbo Pascal- oder Oberon-System;  
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.

Der Prüfungsteilnehmer hat die von ihm erarbeiteten Programme und Module im Quelltext zu kommentieren.

Der Prüfungsteilnehmer hat die von ihm erarbeiteten Quelltexte abzugeben.

## ÖFFNUNG AM 07. MAI 2001

## 1 Problemlösen

- 1.1 Nennen und erläutern Sie drei charakteristische Eigenschaften von Algorithmen!

3 BE

- 1.2 Das Teile-und-herrsche-Verfahren ist ein Verfahren, mit dem Probleme wie folgt gelöst werden:

Das Gesamtproblem wird in Teilprobleme derselben Art wie das Gesamtproblem zerlegt.  
Die Teilprobleme werden dann unabhängig voneinander gelöst.  
Die Lösungen der Teilprobleme werden zu einer Lösung des Gesamtproblems zusammengefügt.

Erläutern Sie, wie das Sortieren durch direktes Mischen als ein Teile-und-herrsche-Verfahren realisiert werden kann!

4 BE

- 1.3 Der Prophet Mohammed verwendete das in Abbildung 1 dargestellte Symbol als Unterschrift.

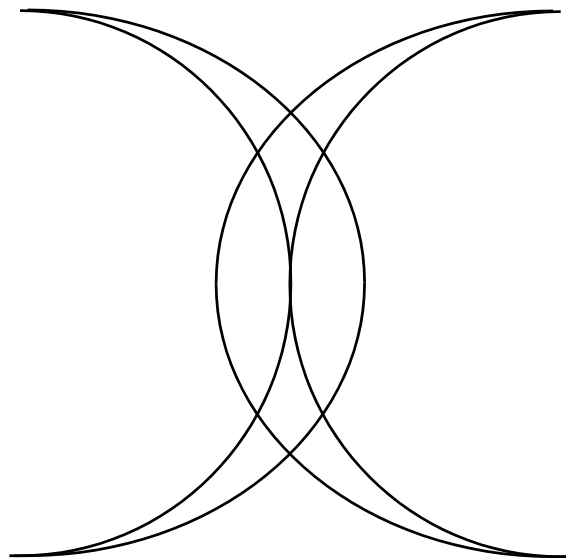


Abbildung 1

Dieses Symbol, das aus Kreisbögen besteht, soll in einem Zug gezeichnet werden. Kein Kreisbogen darf dabei doppelt gezeichnet werden.

Erläutern Sie, wie mit Hilfe der Methode Backtracking ermittelt werden kann, in welcher Reihenfolge die einzelnen Kreisbögen zu zeichnen sind!

5 BE
------

- 1.4 Immer wieder wird die Öffentlichkeit mit Pannen konfrontiert, die auf Programmierfehler zurückzuführen sind. Nennen und erläutern Sie wichtige Methoden der Softwareentwicklung, durch deren Anwendung das Risiko von Programmierfehlern verringert werden kann!

3 BE
------

## 2 UPN-Ausdrücke

Die Postfix-Notation eines Ausdrucks ist auch unter der Bezeichnung umgekehrte polnische Notation (UPN) bekannt.

2.1 Gegeben ist der folgende UPN-Ausdruck:

500 25 90 20 + \* - 300 200 + -

Geben Sie zu diesem UPN-Ausdruck die Baumdarstellung an! Erläutern Sie, wie der Wert dieses UPN-Ausdrucks berechnet wird!

Geben Sie den Wert dieses Ausdrucks an!

5 BE
------

2.2 Entwerfen und implementieren Sie ein Prolog-Programm, das den Wert eines UPN-Ausdrucks berechnet und ausgibt!

Beachten Sie die folgenden Festlegungen:

- Die Operanden sind ganze Zahlen.
- Die Operationen sind Addition, Subtraktion, Multiplikation und ganzzahlige Division.
- Sie können voraussetzen, dass bei der Abfrage ein korrekter UPN-Ausdruck als Liste angegeben wird.
- Eine Abfrage ist zum Beispiel:  
?-wert([ 500, 25, 90, 20, '+', '\*', '-', 300, 200, '+', '-' ], W).

10 BE
-------

### 3.1 Schneller Datenzugriff

Zum schnellen Bereitstellen von Datensätzen zur weiteren Bearbeitung soll ein zweistufiges Speichersystem verwendet werden. Das Speichersystem besteht aus einem großen langsamen Speicher (GLS) und einem kleinen schnellen Speicher (KSS). Alle Datensätze sind dauerhaft in dem Speicher GLS enthalten. Das Bereitstellen eines Datensatzes erfolgt stets aus dem Speicher KSS. Alle Datensätze besitzen die gleiche Länge.

Das Anfordern eines Datensatzes bewirkt in dem Speichersystem die folgenden Vorgänge:

Der angeforderte Datensatz wird aus dem Speicher GLS in den Speicher KSS kopiert, falls er in dem Speicher KSS noch nicht enthalten ist. Soll ein Datensatz in den Speicher KSS kopiert werden und dieser ist voll belegt, so wird zuerst ein Datensatz aus dem Speicher KSS gelöscht. Dann erfolgt das Kopieren und das anschließende Bereitstellen des angeforderten Datensatzes.

Es ist eine Entscheidung zu treffen, welcher Datensatz im Speicher KSS zu löschen ist. Dafür sind mehrere Strategien möglich.

Eine erste Strategie ist das Löschen des Datensatzes, der die längste Zeit nicht angefordert wurde.

Eine zweite Strategie ist das Löschen des Datensatzes, der in einer bestimmten Zeit am wenigsten angefordert wurde.

Eine dritte Strategie ist das Löschen des Datensatzes, der sich die längste Zeit im Speicher KSS befindet.

Wählen Sie für die weitere Arbeit eine der drei Strategien aus!

3.1.1 Erläutern Sie an Hand von Beispielen, wie der Speicher KSS bei Nutzung der von Ihnen ausgewählten Strategie arbeitet!

6 BE
------

3.1.2 Entwerfen Sie ein Programm, das das zweistufige Speichersystem simuliert! Der Speicher KSS soll nach der Strategie arbeiten, die von Ihnen in Teilaufgabe 3.1.1 erläutert wurde.

Implementieren Sie das Programm in Turbo Pascal oder Oberon!

Erläutern Sie, welche Methoden der Softwareentwicklung von Ihnen verwendet wurden!

24 BE
-------

### 3.2 Problem des Handlungsreisenden

Gegeben sind die  $n$  Städte  $s_1, s_2, \dots, s_n$ . Bekannt sind die Entfernungen zwischen den Städten.

Gesucht ist eine kürzeste Rundreise durch die  $n$  Städte. Bei einer Rundreise sind Start- und Zielstadt identisch. Ansonsten ist jede Stadt genau einmal zu besuchen. Eine kürzeste Rundreise durch  $n$  Städte soll mit Hilfe eines rekursiven Algorithmus „Permutation von  $n$  Elementen“ ermittelt werden.

Beispiel: In der folgenden Tabelle sind die Entfernungen zwischen fünf Städten in Kilometern angegeben. Es gilt  $n = 5$ .

	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$
$s_1$	0	70	101	83	52
$s_2$	70	0	59	95	104
$s_3$	101	59	0	69	107
$s_4$	83	95	69	0	53
$s_5$	52	104	107	53	0

Beispiel für eine Rundreise durch fünf Städte: Die Rundreise beginnt in der Stadt  $s_3$ . Dann geht es nach  $s_4, s_1, s_2, s_5$  und wieder nach  $s_3$ .

3.2.1 Entwerfen Sie einen rekursiven Algorithmus zum Ermitteln aller Permutationen von  $n$  Elementen!

6 BE
------

3.2.2 Erläutern Sie den Zusammenhang zwischen der Suche nach einer kürzesten Rundreise durch  $n$  Städte und der Permutation von  $n$  Elementen!

2 BE
------

3.2.3 Entwerfen und implementieren Sie ein Turbo Pascal- oder Oberon-Programm, das  $n$  und die Entfernungen einliest und dann eine kürzeste Rundreise durch die  $n$  Städte ermittelt und ausgibt!

Der rekursive Algorithmus „Permutation von  $n$  Elementen“ ist dabei zu verwenden!

16 BE
-------

3.2.4 Eine kürzeste Rundreise durch  $n$  Städte soll mit Hilfe des von Ihnen in Teilaufgabe 3.2.3 entworfenen und implementierten Programms ermittelt werden. Messen Sie die dafür benötigte Rechenzeit in Abhängigkeit von  $n$ ! Schätzen Sie ab, bis zu welchem  $n$  eine kürzeste Rundreise auf diese Art in zumutbarer Zeit gefunden werden kann!

3 BE
------

3.2.5 Erläutern Sie einen Algorithmus, mit dem eine Näherungslösung für das Problem des Handlungsreisenden gefunden werden kann!

3 BE
------