

# ABITURPRÜFUNG 2006

## 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 10. MAI 2006**

**Aufgabe 1**

- 1.1 Im Jahr 2006 jährt sich der 90. Geburtstag von Claude Shannon. Er sprach erstmals vom Bit und verwendete in diesem Zusammenhang die Symbole 0 und 1. Damit schuf Shannon eine Voraussetzung für das Codieren von Information.

Codieren Sie die Zahl 2006 als Bitfolge.  
Beschreiben Sie das Verfahren, mit dessen Hilfe Sie die Zahl codiert haben.

3 BE
------

- 1.2 Bei einem physikalischen Schülerexperiment werden die Messwerte mit einem Computer erfasst und ausgewertet. Erklären Sie den Informationsfluss von der Aufnahme der Messwerte bis zur Verarbeitung im Computer.

4 BE
------

- 1.3 Setzen Sie sich mit folgender These auseinander:

Nicht Computer, sondern Menschen entscheiden, ob Daten zu Informationen werden.

3 BE
------

- 1.4 Verschlüsseln Sie die folgende Nachricht:

binuebermorgenda

Beschreiben Sie das Verfahren, mit dem Sie die Nachricht verschlüsselt haben.

Kann Ihre Nachricht unbefugt entschlüsselt werden? Begründen Sie Ihre Antwort.

5 BE
------

## Aufgabe 2

Gegeben sind acht Bausteine: ein roter, blauer, gelber, grüner, brauner, weißer, schwarzer und grauer. Mit diesen Bausteinen soll nach folgenden Regeln ein Turm gebaut werden.

1. Regel: Der Turm muss aus den acht Bausteinen bestehen.
2. Regel: Unmittelbar auf jeden Baustein darf höchstens ein Baustein gelegt werden:
  - auf den roten der blaue oder schwarze,
  - auf den blauen der gelbe, braune oder rote,
  - auf den gelben der grüne oder weiße,
  - auf den grünen der rote, gelbe oder braune,
  - auf den braunen der graue,
  - auf den weißen der graue,
  - auf den schwarzen der grüne oder weiße und
  - auf den grauen der blaue oder schwarze.

Ein Turm kann den roten Baustein als Sockel haben. Geben Sie die Bausteine an, die ebenfalls Sockel sein können.

Entwerfen und implementieren Sie ein Programm in Prolog, das bei Anfrage alle verschiedenen Möglichkeiten zum Bauen eines Turms ermittelt und ausgibt. Die Ausgabe soll mit dem Sockelbaustein beginnen und die richtige Reihenfolge der Bausteine angeben.

Geben Sie zu Ihrem Programm eine entsprechende Anfrage an, so dass alle verschiedenen Möglichkeiten zum Bauen eines Turms ermittelt und ausgegeben werden.

15 BE
-------

### Aufgabe 3.1

Beschreiben Sie einen Algorithmus, der ein gegebenes Wort in einem Text sucht.

Erklären Sie den Begriff „Zeitkomplexität eines Algorithmus“. Beurteilen Sie die Zeitkomplexität des von Ihnen beschriebenen Algorithmus.

Gegeben ist eine Tabelle mit acht Zeilen und zehn Spalten. In der Tabelle ist in den Zeilen und Spalten u. a. das Wort HUND viermal, KATZE zweimal und MAUS dreimal enthalten.

R	X	O	J	K	H	C	K	U	H
S	K	I	M	A	U	S	U	H	U
B	N	H	A	T	N	C	H	U	N
R	H	M	U	Z	D	H	U	N	D
M	A	U	S	E	K	A	T	Z	E
A	N	H	E	N	U	F	E	T	T
U	D	I	N	O	H	U	N	D	E
D	N	U	H	T	E	R	G	L	Z

Beachten Sie den Hinweis:

Die Zeilen sind von links nach rechts und die Spalten von oben nach unten zu lesen.

Entwerfen und implementieren Sie ein Programm in Pascal oder Oberon, das Folgendes leistet:

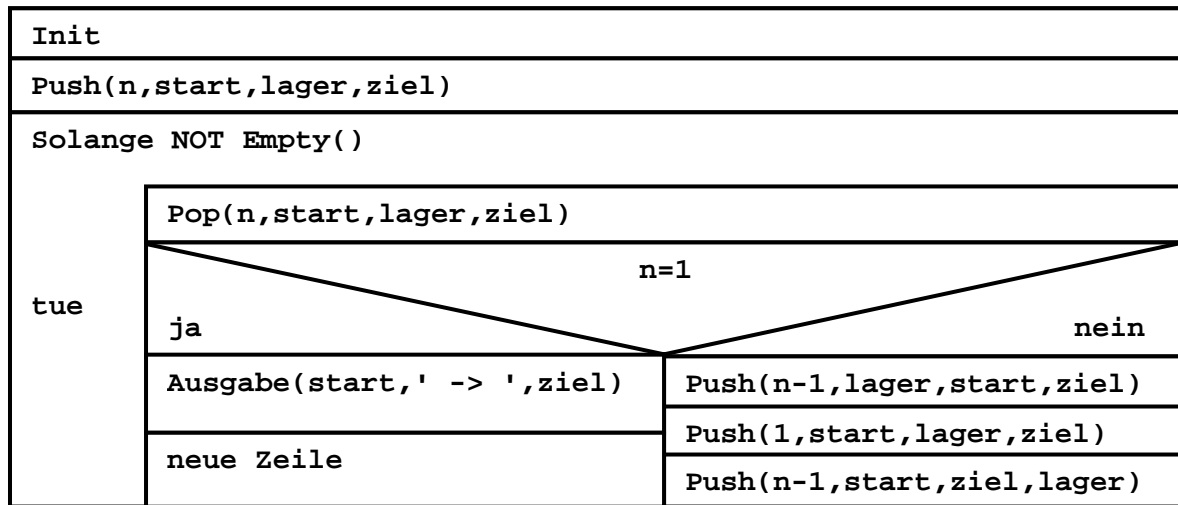
- Einlesen einer Tabelle mit maximal 20 Zeilen und maximal 20 Spalten,
- Einlesen eines Wortes,
- Ermitteln der Anzahl des Vorkommens dieses Wortes in der Tabelle,
- Ausgabe der Anzahl.

Testen Sie das Programm mit Hilfe der gegebenen Tabelle und den Wörtern HUND, KATZE und MAUS. Dokumentieren Sie die Tests.

30 BE
-------

### Aufgabe 3.2

Der im Struktogramm dargestellte Algorithmus löst das Problem der Türme von Hanoi iterativ:



Im Algorithmus ist  $n$  eine ganze Zahl ( $1 \leq n \leq 10$ ) und start, lager sowie ziel ist jeweils ein Zeichen. Init, Push, Pop und Empty sind Operationen eines Stapels. Jedes Element des Stapels besteht aus vier Komponenten, und zwar aus einer ganzen Zahl und drei Zeichen. Die Operationen des Stapels leisten Folgendes:

- Init erzeugt den leeren Stapel.
- Push(z,c1,c2,c3) stapelt ein Element. Die Werte z (z ist eine ganze Zahl) und c1,c2 sowie c3 (c1, c2, c3 sind Zeichen) sind Komponenten des Elements.
- Pop(z,c1,c2,c3) liefert die Werte z, c1, c2 sowie c3 der Komponenten des obersten Elements und entstapelt das Element.
- Empty() prüft, ob der Stapel leer ist.

Testen Sie den Algorithmus mit den Werten  $n = 3$ , start = 'S', lager = 'L' und ziel = 'Z'. Dokumentieren Sie den Test.

Erläutern Sie die Arbeitsweise eines Stapels.

Entwerfen Sie ein Modul `stapel`, das den oben beschriebenen Stapel realisiert. Implementieren Sie das Modul in Pascal oder Oberon.

Implementieren Sie den im Struktogramm dargestellten Algorithmus in einem Pascal- oder Oberon-Programm. Das Programm soll das Modul `stapel` importieren und verwenden.

Entwerfen Sie einen Algorithmus, der das Problem der Türme von Hanoi rekursiv löst. Implementieren Sie den Algorithmus in einem Pascal- oder Oberon-Programm.