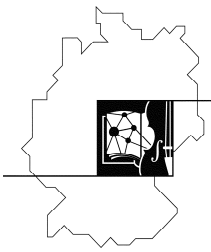

FREISTAAT THÜRINGEN

Thüringer Ministerium für Bildung,
Wissenschaft und Kultur

ABITURPRÜFUNG 2010

LEISTUNGSFACH

INFORMATIK

(HAUPTTERMIN)

Bearbeitungszeit: 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 28. APRIL 2010

Aufgabe 1

Am 22. Juni 2010 jährt sich der 100. Geburtstag von Konrad Zuse. Er entwickelte und realisierte die Gleitkommadarstellung der reellen Zahlen.
Dazu verwendete er die reellen Zahlen in normalisierter Form. Im Dezimalsystem bedeutet das, dass jede reelle Zahl mit einer Ziffer vor dem Komma, Ziffern nach dem Komma und einer Zehnerpotenz angegeben wird. Die Ziffer vor dem Komma ist verschieden von Null.

Beispiele:

Zahl	normalisierte Darstellung
234,5	$2{,}345 \cdot 10^{2}$
0,00321	$3{,}21 \cdot 10^{-3}$

Die Ziffern der reellen Zahl speichert man in der Mantisse. Der Exponent wird ebenfalls gespeichert.

Im Folgenden werden nur noch positive reelle Zahlen und deren Gleitkommadarstellungen mit vier Stellen in der Mantisse und zwei Stellen im Exponenten betrachtet.

Beispiel	Exponent			Mantisse			
234,56		0	2	2	3	4	5
0,00321	-	0	3	3	2	1	0
47,32169		0	1	4	7	3	2

Ermitteln Sie die kleinste und die größte reelle Zahl, die auf diese Weise dargestellt werden kann.

Gegeben sind zwei Algorithmen:

a)

v := 234,56
v := v – 0,0000321
Ausgabe v

b)

u := 20 / 6
u := u * 12
Ausgabe u

Geben Sie für jeden Algorithmus die reelle Zahl an, die ausgegeben wird, wenn mit der oben beschriebenen Gleitkommadarstellung gearbeitet wird.

Vergleichen Sie die Menge der reellen Zahlen mit der Menge der in Gleitkommadarstellung verwendeten Zahlen im Rechner.

Der Plankalkül war ein Vorläufer imperativer Programmiersprachen und wurde von Konrad Zuse entwickelt.
Erläutern Sie das imperative Sprachparadigma.

Zuses Programmiersprache umfasst auch zusammengesetzte Datentypen.
Beschreiben Sie zwei zusammengesetzte Datentypen in einer Ihnen bekannten Programmiersprache und deren wesentliche Operationen und Relationen.

Setzen Sie sich mit folgendem Zitat von Konrad Zuse auseinander.
„Die Gefahr, dass der Computer so wird wie der Mensch, ist nicht so groß wie die Gefahr, dass der Mensch so wird wie der Computer.“

15 BE

Aufgabe 2

Der Schriftsteller Thomas Mann erzählt im Roman „Buddenbrooks“ die Geschichte einer hanseatischen Kaufmannsfamilie. Im nachstehenden Text ist ein Ausschnitt aus dem Stammbaum dieser Familie gegeben:

Antoinette, Klothilde, Josephine, Tony, Clara, Friederike, Olly, Erika, Henriette, Bethsy, Elisabeth und Pfiffi sind die weiblichen Familienmitglieder.
Johan, Johann, Bernhard, Gotthold, Thomas, Christian, Jean und Hanno sind die männlichen Familienmitglieder.
Johann und Bernhard sind die Kinder von Johan.
Gotthold und Jean sind die Kinder von Johann.
Gotthold ist das Kind von Josephine.
Klothilde ist das Kind von Bernhard.
Jean und Olly sind die Kinder von Antoinette.
Thomas, Tony, Christian und Clara sind die Kinder von Jean.
Elisabeth ist das Kind von Erika.
Friederike, Henriette und Pfiffi sind die Kinder von Gotthold.
Clara, Thomas, Christian und Tony sind die Kinder von Bethsy.
Hanno ist das Kind von Thomas.
Erika ist das Kind von Tony.

Übertragen Sie die Fakten aus dem Text in ein Prolog-Programm.

Geben Sie zum Programm die Anfragen an:

- Ist Christian ein Kind von Jean?
- Ist Pfiffi ein weibliches Familienmitglied?
- Ist Hanno ein männliches Familienmitglied?
- Wessen Kind ist Henriette?
- Wer sind die Kinder von Gotthold?
- Wer ist wessen Kind?
- Ist Tony ein männliches Familienmitglied?

Geben Sie an, welche Antwort das Prolog-System auf die letzte Anfrage ausgibt. Begründen Sie, weshalb das Prolog-System diese Antwort ausgibt.

Erweitern Sie das Programm um die Prädikate:

- Elternteil-Kind-Beziehung
- Vater-Sohn-Beziehung
- Mutter-Tochter-Beziehung
- Geschwister-Beziehung
- Halbgeschwister-Beziehung
- Großmutter-Enkel-Beziehung
- Urgroßvater-Urenkel-Beziehung

Geben Sie zum Programm die Anfragen an:

- Ist Johan ein Elternteil von Bernhard?
- Wessen Großmutter ist Antoinette?
- Wer sind Tonys Geschwister?
- Wessen Urgroßvater ist Johan?
- Ist Hanno der Sohn von Thomas?
- Wer sind wessen Halbgeschwister?

Erläutern Sie, wie das Prolog-System alle Antworten auf die letzte Anfrage sucht.

Erweitern Sie das Programm, so dass die Frage beantwortet wird:
Wer ist wessen Mutter?

Erweitern Sie das Programm um ein rekursives Prädikat, das bei Anfrage die Vorfahren eines Familienmitglieds ermittelt.
Geben Sie zum Programm die Anfrage an:
Wer sind die Vorfahren von Elisabeth?

15 BE

Aufgabe 3.1

Positive ganze Zahlen sollen in einer Hashtabelle abgelegt werden.
Dazu wird jede positive ganze Zahl x in einer Zeile der Hashtabelle gespeichert.
Die Hashtabelle hat p Zeilen.
Mithilfe der Hashfunktion h wird für die positive ganze Zahl x eine Adresse berechnet. Die Adresse entspricht der Zeilennummer der Hashtabelle.

Die Hashfunktion h lautet:
h(x) = x MOD p
Die Operation MOD berechnet den Rest bei der ganzzahligen Division.
Die Anzahl der Zeilen p der Hashtabelle muss eine Primzahl sein.

Beispiel für Berechnungen einiger Adressen einer Hashtabelle mit 7 Zeilen:

h(23657) = 23657 MOD 7 = 4
h(118712) = 118712 MOD 7 = 6
h(63) = 63 MOD 7 = 0
h(4212) = 4212 MOD 7 = 5
h(412) = 412 MOD 7 = 6

Bei der Hashfunktion h kann es vorkommen, dass zwei positive ganze Zahlen die gleiche Adresse erhalten, zum Beispiel 118712 und 412. Mehrere positive ganze Zahlen mit gleicher Adresse werden als absteigend sortierte Liste in der Hashtabelle an die Adresse angefügt.
In der Abbildung 3.1 ist eine Hashtabelle mit sieben Adressen und den gespeicherten positiven ganzen Zahlen angegeben.

0	456981	1841	868	322
1	6483			
2				
3	622219	3986		
4	23657			
5	4212			
6	456987	118712	412	

Abbildung 3.1

Entwerfen und implementieren Sie ein Programm, das positive ganze Zahlen in eine Hashtabelle speichert.

Beachten Sie folgende Hinweise:
- Die Anzahl der positiven ganzen Zahlen ist mindestens 100.
- Jede positive ganze Zahl wird von einem Zufallsgenerator erzeugt.

- Die Hashtabelle hat mindestens fünf Zeilen. Die Adressierung erfolgt in fortlaufender Nummerierung beginnend mit 0.
- Das Programm gibt die Adressen und die zugeordneten positiven ganzen Zahlen wie in der Abbildung 3.1 dargestellt aus.

Erweitern Sie Ihr Programm so, dass positive ganze Zahlen aus der Hashtabelle gelöscht werden können.

Testen Sie, ob Ihr Programm die Aufgabenstellungen löst.
Erläutern Sie, wie Sie den Test durchgeführt haben.

Vergleichen Sie die Suche in einer Hashtabelle mit der Suche in einer Liste.

30 BE

Aufgabe 3.2

Eine Person verschlüsselt einen Klartext mithilfe des folgenden Verfahrens:

- Zur Erzeugung des Schlüssels wählt die Person eine ganze Zahl z ($2 \leq z \leq 6$) aus. Die ganzen Zahlen von 1 bis z werden permutiert. Davon wird eine Permutation ausgewählt. Das ist der Schlüssel.
- In eine Tabelle mit z Spalten wird jede Zahl der Permutation von links nach rechts in genau eine Spalte der ersten Zeile geschrieben.
- Von der zweiten Zeile an wird jeder Buchstabe des Klartexts fortlaufend von links nach rechts in genau eine Spalte geschrieben.
- In der letzten Zeile dürfen sich links bis zur Spalte mit dem letzten Buchstaben keine leeren Spalten befinden. Ist die Bedingung nicht erfüllt, wird in jede dieser leeren Spalten das Nummernzeichen # geschrieben.
- Die Spalten werden so umgeordnet, dass die Zahlen in der ersten Zeile in aufsteigender Reihenfolge stehen.
- Von der zweiten Zeile an werden die Buchstaben aus den Spalten von links nach rechts gelesen und zum Geheimtext zusammengefügt.

Ein Beispiel:

<table>
<tr><td>Klartext:</td><td>abiturpruefungstag</td></tr>
<tr><td>Ganze Zahl:</td><td>5</td></tr>
<tr><td>Schlüssel:</td><td>5 4 2 3 1</td></tr>
<tr><td>Tabelle mit Klartext:</td><td>
<table>
<tr><td>5</td><td>4</td><td>2</td><td>3</td><td>1</td></tr>
<tr><td>a</td><td>b</td><td>i</td><td>t</td><td>u</td></tr>
<tr><td>r</td><td>p</td><td>r</td><td>u</td><td>e</td></tr>
<tr><td>f</td><td>u</td><td>n</td><td>g</td><td>s</td></tr>
<tr><td>t</td><td>a</td><td>g</td><td>#</td><td>#</td></tr>
</table>
</td></tr>
<tr><td>Tabelle mit Geheimtext:</td><td>
<table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr>
<tr><td>u</td><td>i</td><td>t</td><td>b</td><td>a</td></tr>
<tr><td>e</td><td>r</td><td>u</td><td>p</td><td>r</td></tr>
<tr><td>s</td><td>n</td><td>g</td><td>u</td><td>f</td></tr>
<tr><td>#</td><td>g</td><td>#</td><td>a</td><td>t</td></tr>
</table>
</td></tr>
<tr><td>Geheimtext:</td><td>uitbaeruprsnguf#g#at</td></tr>
</table>

Gegeben ist der Geheimtext `aidsttssegrneeghm#i#` und der Schlüssel 3 1 4 2.
Beschreiben Sie das Entschlüsseln des Geheimtexts.
Geben Sie den Klartext an.
Geben Sie einen Algorithmus an, der für eine ganze Zahl n ($n \geq 2$) alle Permutationen von 1 bis n ermittelt.

Entwerfen und implementieren Sie ein Programm, das Folgendes leistet:
- Einlesen eines Klartexts
- Auswahl einer ganze Zahl z ($2 \leq z \leq 6$) mithilfe des Zufallsgenerators
- Ermitteln aller Permutationen der ganzen Zahlen von 1 bis z
- Auswahl einer Permutation mithilfe des Zufallsgenerators
- Verschlüsseln des Klartexts unter Anwendung des beschriebenen Verfahrens
- Ausgabe des Geheimtexts

Beachten Sie die Festlegung, dass im Programm der Algorithmus zum Ermitteln der Permutationen der ganzen Zahlen von 1 bis n zu verwenden ist.

Erweitern Sie das Programm, so dass es zusätzlich Folgendes leistet:
- Einlesen eines Geheimtexts und der zugehörigen Permutation
- Überführen des Geheimtexts in den Klartext unter Anwendung des beschriebenen Verfahrens
- Ausgabe des Klartexts

Testen Sie, ob das Programm die Aufgabenstellungen löst.
Beschreiben Sie den Test Ihres Programms.

Erläutern Sie die Methoden der Softwareentwicklung, die von Ihnen verwendet wurden.

30 BE