

**Die Aufgaben umfassen 7 Seiten****Aufgabe 1:****1.1 Iteration und Rekursion**

Die Multiplikation zweier natürlicher Zahlen ( $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ ) kann durch eine fortgesetzte Addition realisiert werden. So ist z. B.  $3 \cdot 7 = 7 + 7 + 7$ .

1.1.1 Schreiben Sie eine Delphi (Pascal)-Funktion bzw. Java-Methode `mult1(a, b)`, die das Produkt zweier natürlicher Zahlen  $a$  und  $b$  durch fortgesetzte Addition iterativ bestimmt.

1.1.2 Der Vorgang der fortgesetzten Addition kann auch rekursiv formuliert werden:

$$(*) \quad a \cdot b = \begin{cases} b + (a-1) \cdot b & \text{für } a > 0 \\ 0 & \text{sonst} \end{cases}$$

a) Berechnen Sie schriftlich nach dieser rekursiven Vorschrift das Produkt  $3 \cdot 7$  unter Angabe aller Rechenschritte

b) Schreiben Sie eine Delphi (Pascal)-Funktion bzw. Java-Methode `mult2(a, b)`, die das Produkt zweier natürlichen Zahlen  $a$  und  $b$  nach dieser Vorschrift berechnet.

1.1.3 Betrachten Sie jetzt den Fall, dass die Faktoren **ganze** Zahlen sein dürfen. Schreiben Sie dazu eine Delphi (Pascal)-Funktion bzw. Java-Methode `mult3(a, b)`, die unter Verwendung von `mult1` oder `mult2` das Produkt zweier **ganzer** Zahlen  $a$  und  $b$  berechnet.

Fach: Informatik

Prüfungsart: G-Niveau

Dauer: 3 Stunden

Hilfsmittel: Taschenrechner

## 1.2 ADT Liste

Gegeben ist die folgende einfach verkettete Liste von ganzzahligen Werten:



- 1.2.1 Geben Sie die Deklaration bzw. Klassendefinition des Knotentyps an. Verwenden Sie dabei die Bezeichner `data` bzw. `next` für die Komponenten des Knotentyps



- 1.2.2 Im Folgenden sollen elementare Operationen an der Liste L durchgeführt werden. Geben Sie die dafür erforderlichen Anweisungen für die Operationen in Delphi oder Java an.

- Entfernen Sie den 1. Knoten aus der Liste L.
- Fügen Sie nach dem 1. Knoten der Liste L einen neuen Knoten mit dem Inhalt 45 in die Liste L ein.

- 1.2.3 Implementieren Sie eine Delphi (Pascal)-Funktion bzw. Java-Methode, die die Summe aller Inhalte der Knoten der Liste L berechnet.

Fach: Informatik

Prüfungsart: G-Niveau

Dauer: 3 Stunden

Hilfsmittel: Taschenrechner

**Aufgabe 2:****2.1 Sprachen und reguläre Ausdrücke**

Gegeben ist die Sprache  $L = \{10^i 1^j 0 \mid \text{mit } i, j \geq 0 \text{ und } i, j \text{ gerade}\}$  über dem Alphabet  $\{0,1\}$ .

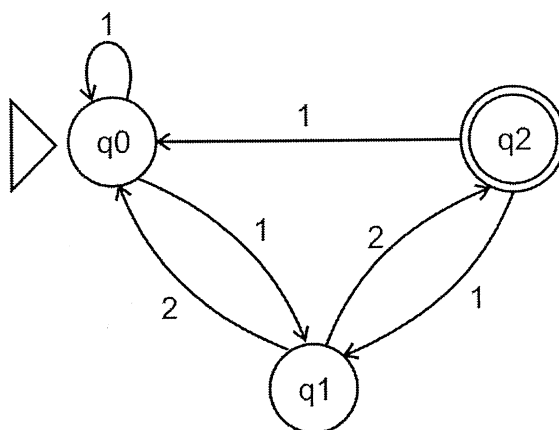
2.1.1 Geben Sie einen regulären Ausdruck für die Sprache L an.

2.1.2 Geben Sie eine Grammatik G an, die die Sprache L erzeugt

2.1.3 Konstruieren Sie den Übergangsgraphen eines endlichen Automaten, der die Sprache L akzeptiert, wobei der Automat nicht unbedingt deterministisch sein muss.

**2.2 Automaten**

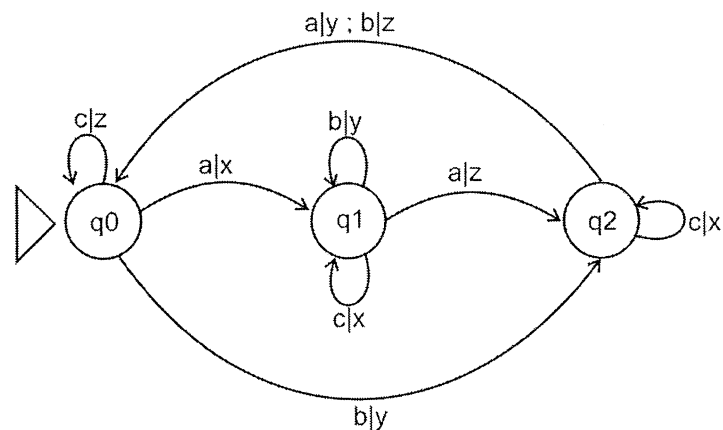
Gegeben ist der Übergangsgraph eines nichtdeterministischen Automaten.



Erzeugen Sie mit der Teilmengenkonstruktion einen deterministischen Automaten, der die gleiche Sprache wie der Automat akzeptiert.

### 2.3 Anwendung eines endlichen Automaten

Gegeben ist der Übergangsgraph eines endlichen Automaten A mit Ausgabe



Der Automat kann z. B. dazu dienen, ein Wort bestehend aus den Zeichen  $a, b, c$  in ein Wort  $w^*$  bestehend aus den Zeichen  $x, y, z$  zu übersetzen.

- 2.3.1 Geben Sie die Komponenten des Automaten an.
- 2.3.2 Übersetzen Sie mit diesem Automaten das Wort  $w=abcbabc$ . Geben Sie die Zustände an, die der Automat dabei durchläuft.
- 2.3.3 Der Automat erzeugt die Ausgabe  $w^*=zx yzx y$ . Bestimmen Sie das Eingabewort  $w$  und beschreiben Sie Ihre Vorgehensweise.

Fach: Informatik

Prüfungsart: G-Niveau

Dauer: 3 Stunden

Hilfsmittel: Taschenrechner

**Aufgabe 3:****3.1 Sortierv Verfahren Maxsort und Bubblesort**

3.1.1 Gegeben ist eine zufällige Anordnung von Zahlen, die nach einem vorgegebenen Verfahren sortiert werden soll. Das hier dargestellte Sortierv Verfahren heißt *Maxsort*.

Beispiel:

	1.	2.	3.	4.	5.	6.	7.
Ausgangsanzordnung	40	4	38	13	2	23	8
Nach dem 1. Durchlauf	8	4	38	13	2	23	<b>40</b>
Nach dem 2. Durchlauf	8	4	23	13	2	<b>38</b>	<b>40</b>
Nach dem 3. Durchlauf	8	4	2	13	<b>23</b>	<b>38</b>	<b>40</b>
Nach dem 4. Durchlauf	8	4	2	<b>13</b>	<b>23</b>	<b>38</b>	<b>40</b>
Nach dem 5. Durchlauf	2	4	<b>8</b>	<b>13</b>	<b>23</b>	<b>38</b>	<b>40</b>
Nach dem 6. Durchlauf	2	<b>4</b>	<b>8</b>	<b>13</b>	<b>23</b>	<b>38</b>	<b>40</b>

Beschreiben Sie das Maxsort-Verfahren allgemein für n Zahlen.

3.1.2 Sortieren Sie die oben angegebene Ausgangsanzordnung nach dem Bubblesort-Verfahren.

Stellen Sie wie oben den Zustand nach jedem Durchlauf in der vorgegebenen Tabelle auf dem Blatt dar.

*Hinweise:*

- Nach jedem Durchlauf soll eine weitere Zahl richtig eingeordnet sein.
- Die vorgegebene Tabelle enthält unter Umständen mehr Zeilen als notwendig sind. Entwerten Sie die nicht benötigten Zeilen.
- Zwischenschritte können auf einem gesonderten Blatt notiert werden.

	1.	2.	3.	4.	5.	6.	7.
Ausgangsanzordnung	40	4	38	13	2	23	8
Nach dem 1. Durchlauf							
Nach dem 2. Durchlauf							
Nach dem 3. Durchlauf							
Nach dem 4. Durchlauf							
Nach dem 5. Durchlauf							
Nach dem 6. Durchlauf							

Fach: Informatik

Prüfungsart: G-Niveau

Dauer: 3 Stunden

Hilfsmittel: Taschenrechner

### **3.2 Laufzeit des Maxsort-Verfahrens und Vergleich mit Bubblesort-Verfahren**

*Hinweis:* Bei der Beurteilung des Laufzeitverhaltens eines Verfahrens sollen nur die notwendigen Vergleiche von Elementen herangezogen werden.

#### **3.2.1 Laufzeit des Maxsort-Verfahrens**

- a) Ermitteln Sie ganz konkret die Gesamtanzahl der notwendigen Vergleiche aus Aufgabe 3.1.1 nach dem Maxsort-Verfahren.
- b) Bestimmen Sie das Laufzeitverhalten für eine große Anzahl von  $n$  Elementen und geben Sie dieses in der O-Notation an.

#### **3.2.2 Laufzeit des Bubblesort-Verfahrens und Vergleich mit dem Maxsort-Verfahren**

- a) Geben Sie das Laufzeitverhalten des Bubblesort-Verfahrens im günstigsten und im ungünstigsten Fall an. Begründen Sie Ihre Aussage.
- b) Vergleichen Sie das Laufzeitverhalten des Bubblesort-Verfahrens mit dem Laufzeitverhalten des Maxsort-Verfahrens. Betrachten Sie dabei speziell die Bedeutung der Vorsortierung auf das Laufzeitverhalten bei beiden Verfahren.

Fach: Informatik

Prüfungsart: G-Niveau

Dauer: 3 Stunden

Hilfsmittel: Taschenrechner

### 3.3 Substitutionsverfahren

#### 3.3.1 Angriffe auf Substitutionsverfahren

In der deutschen Sprache treten bestimmte Zeichen häufiger auf als andere, z.B. das „e“ mit einer Häufigkeit von ca. 17%. Die Häufigkeitsverteilung der Zeichen lässt sich bei bestimmten Substitutionsverfahren dazu verwenden, um Rückschlüsse auf den Klartext zu ziehen.

- a) Beim folgenden Substitutionsverfahren ist das Geheimalphabet durch Permutation (Vertauschen) des Klartextalphabetes entstanden. Die folgende Tabelle zeigt einen Ausschnitt:

Klarzeichen	a	b	c	d	e	...	x	y	z
Geheimzeichen	z	e	u	a	g	...	x	i	m

Beschreiben Sie, wie die Kryptoanalyse eines natürlichsprachlichen, genügend langen Textes bei diesem Substitutionsverfahren durchgeführt wird.

- b) Welche Eigenschaft müssen Substitutionsverfahren allgemein besitzen, damit eine Kryptoanalyse mittels Häufigkeitsanalyse erfolgsversprechend ist?
- c) Begründen Sie, dass das RSA-Verfahren ebenfalls ein Substitutionsverfahren darstellt und erläutern Sie, durch welche Methode der Angriff durch Häufigkeitsanalyse bei dem RSA-Verfahren erschwert wird.

#### 3.3.2 Anwendung

Man kann einen endlichen Automaten wie in Aufgabe 2.3 ohne Probleme dahingehend erweitern, dass er alle Buchstaben verarbeiten kann.

- a) Begründen Sie, dass ein solcher Automat ein Substitutionsverfahren darstellt.
- b) Warum ist die Kryptoanalyse mittels der Häufigkeitsverteilung hier erschwert?