

**Aufgabe 1.1** (6 Punkte)

Eine einfach-verkettete Liste sei durch die folgenden Arrays gegeben.

	A	NEXT
0	–	1
1	4	2
2	10	4
3	7	6
4	9	5
5	6	3
6	2	0
7		

- Stellen Sie die Zeigerstruktur der Liste dar.
- Geben Sie die Arrays nach Aufruf der Prozedur `INSERT(Obj, p, q)` mit `Obj=12`, `p=4` und `q=7` an.
- Geben Sie die Arrays an, wenn nachträglich der Aufruf von `DELETE(1)` erfolgte.

**Aufgabe 1.2** (6 Punkte)

Entwerfen Sie für die Array-Implementierung einer zweifach-verketteten Liste die Prozedur `DELETE(p)`, die das Element nach dem Element mit Index  $p$  aus der Liste entfernen soll. Berücksichtigen Sie insbesondere die Entfernung des ersten oder letzten Elements der Liste.

**Aufgabe 1.3** (6 Punkte)

Gegeben seien drei Stacks  $S_1, S_2, S_3$  durch die Arrays  $A_1, A_2, A_3$  und die Variablen  $TOP_1, TOP_2$  und  $TOP_3$ . Alle drei Stacks können gerade und ungerade Zahlen aufnehmen.  $S_1$  und  $S_2$  sollen bereits Zahlen enthalten,  $S_3$  sei zunächst leer. In dieser Aufgabe dürfen die Prozeduren `INSERT` und `DELETE` für Stacks verwendet werden.

- Schreiben Sie eine Prozedur, nach deren Ausführung alle geraden Zahlen auf  $S_1$  und alle ungeraden Zahlen auf  $S_2$  liegen. Dabei sollen keine weiteren Stacks außer  $S_1, S_2$  und  $S_3$  benutzt werden.
- Die Anzahl der geraden Zahlen auf dem Stack  $S_1$  sei  $k_1$ , die der ungeraden Zahlen  $l_1$ . Analog sei  $k_2$  die Anzahl der geraden Zahlen und  $l_2$  die Anzahl der ungeraden Zahlen auf  $S_2$ . Geben Sie für Ihre Lösung aus a) die Anzahl der Umschichtungen von Zahlen in Abhängigkeit von  $k_1, k_2, l_1$  und  $l_2$  an, ein Aufruf von `INSERT` mit anschließendem `DELETE` zählt dabei als eine Umschichtung.

**Aufgabe 1.4** (6 Punkte)

In der Vorlesung wurde die Datenstruktur *einfach-verkettete Liste* definiert. Auf Grundlage dieser Datenstruktur definieren wir eine neue Datenstruktur *zirkuläre Liste*, für die der letzte „echte“ Knoten auf einen vorhergehenden Knoten zeigt. Hier ein Beispiel:

	A	NEXT
0	–	1
1	4	2
2	2	3
3	0	4
4	9	5
5	7	6
6	6	7
7	8	8
8	1	5

- Geben Sie einen Algorithmus an, der prüft, ob die übergebene verkettete Liste zirkulär ist. Dieser darf die Arrays A und NEXT nicht verändern und darüber hinaus nur einen Platzbedarf von  $O(1)$  haben. Der Zeitbedarf muss  $O(n)$  sein (mit  $n = \text{Anzahl der Knoten}$ ). Sie dürfen voraussetzen, dass alle Werte im Array A unterschiedlich sind.  
Vorsicht: Der Algorithmus kennt *nicht* die Anzahl  $n$  der Knoten, so dass es nicht ausreicht zu testen, ob man nach  $n$  Schritten das Ende der Liste erreicht.
- Skizzieren Sie einen Korrektheitsbeweis Ihres Algorithmus.
- Zeigen Sie, dass Ihr Algorithmus den vorgegebenen Platz- und Zeitbedarf hat.

**VSPL-Anmeldung:** Bitte melden Sie sich vor dem Abgabeschluss am 17.04. über VSPL für die Übungen zu Datenstrukturen (150323) an.

---

**Abgabe:** Lösungen können jeweils bis zum folgenden Dienstag um 12:00 Uhr in die Kästen vor NA 02/257 (Nähe Rechenzentrum Servicecenter) *nach Aufgaben getrennt* eingeworfen werden. Geben Sie Ihren Namen, Ihre Matrikelnummer und Ihre Gruppe an. Auf jedem abgegebenen Aufgabenzettel dürfen bis zu drei Namen stehen.