

## **Aufgabe 1: Helden der Informatik**

Benennen Sie mindestens einen Beitrag, der den jeweiligen Menschen zur wichtigen Person der Informatik macht:

Alan Turing:

John von Neumann:

Konrad Zuse:

Name: \_\_\_\_\_

Matr.-Nr: \_\_\_\_\_

## Aufgabe 2: Grammatiken

Gegeben sei eine Grammatik wie folgt ( $\varepsilon$  ist Leersymbol):

G = <Startsymbol, Terminale, Nicht-Terminale, Regeln> mit

Startsymbol = *Start*

Terminale = {*k, a, i, b*}

Nicht-Terminale = {*X, Y, Z*}

Regeln = {*Start*  $\rightarrow$  *X*,  
*Start*  $\rightarrow$  *YX*,  
*Y*  $\rightarrow$  *k*,  
*Y*  $\rightarrow$  *a*,  
*Y*  $\rightarrow$  *i*,  
*X*  $\rightarrow$  *Z*,  
*X*  $\rightarrow$  *ZX*,  
*Z*  $\rightarrow$  *ab*,  
*Z*  $\rightarrow$  *ba* }

Entscheiden Sie, ob folgende Wörter in der durch die Grammatik definierten Sprache enthalten sind:

- |                 |           |
|-----------------|-----------|
| • $\varepsilon$ | ja / nein |
| • abba          | ja / nein |
| • acdc          | ja / nein |
| • kabba         | ja / nein |
| • aabbaa        | ja / nein |
| • abbaabba      | ja / nein |
| • ababccdd      | ja / nein |
| • ibaba         | ja / nein |
| • iabiba        | ja / nein |
| • ababab        | ja / nein |

Bestimmen Sie die Wörter, die durch die Grammatik akzeptiert werden in der Art  $a^n b^n$ :

Name: \_\_\_\_\_

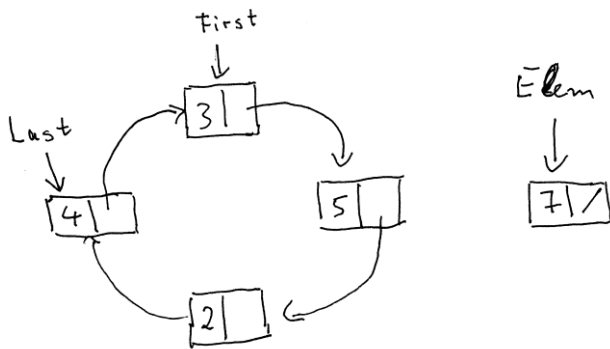
Matr.-Nr: \_\_\_\_\_

## Aufgabe 3: Ring

Gegeben sei eine Implementation eines Ringes (das letzte Element zeigt wieder auf das erste Element), realisiert als einfach verkettete Liste mit den Zeigern `First` auf das erste Element und `Last` auf das letzte Element. Skizzieren Sie eine Funktion `Push` als Pseudocode, die ein Element `Elem` hinter `Last` einfügt (Sonderfall Ring ist leer beachten) sowie eine Funktion `Weiter`, die das nächste Element des Rings liefert ohne Entfernen aus dem Ring und dabei `First` und `Last` updated.

```
struct Elem {integer val; ^Elem next}
```

Beispiel eines Ringes und neuen Elements:



Name: \_\_\_\_\_

Matr.-Nr: \_\_\_\_\_

## Aufgabe 4: Prolog Unifikation

Gegeben seien jeweils 2 Prolog-Ausdrücke. Sie sollen entscheiden, ob diese beiden Ausdrücke unifizierbar sind und, falls dieses der Fall ist, den durch Substitution erzeugten Ausdruck angeben. Sollte die Unifikation misslingen, geben Sie den Grund an. Als Beispiele sind gegeben (Konstanten fangen mit kleinem, Variablen mit großem Buchstaben an):

- `test (X, foo, Y)`    =    `test (M, N, M)`    ☺    mit `test (X, foo, X)`.
- `test (kai, X)`        =    `test1(kai, udo)`    ☹    weil `test ≠ test1`.

Hier nun die Ausdrücke:

- `person(FN, vonLuck)`    =    `person(kai, LN)`.
- `vorname(Kai)`            =    `nachname(vonLuck)`.
- `append([1,2],X,X)`       =    `append(X,[1,2],[1,2])`.
- `name(Kai)`                =    `name(vonLuck)`.

Name: \_\_\_\_\_

Matr.-Nr: \_\_\_\_\_

## Aufgabe 5: Endlicher Automat

Gegeben sei ein Getränkeautomat, der Wasser zu 40 Cent und Brause zu 50 Cent anbietet. Die Auswahl Wasser oder Brause erfolgt am Anfang durch einen Auswahlbutton. Der Automat akzeptiert 20 Cent und 50 Cent Münzen, er wechselt durch Ausgabe von 10 Cent und 20 Cent Münzen. Skizzieren Sie den entsprechenden endlichen Automaten.

Name: \_\_\_\_\_

Matr.-Nr: \_\_\_\_\_