Aufgabe 1: Dank Informatik

Benennen Sie die drei Techniken / Fähigkeiten, die dank Informatik kaum noch benötigt werden. Als Beispiel sei gegeben:

Technikablösung:

Rechenschieber / Fähigkeiten, ihn zu bedienen ersetzt durch Taschenrechner

Technikablösung 1:

Technikablösung 2:

Technikablösung 3:

Name:_____ Matr.-Nr: _____

Aufgabe 2: Grammatiken

Gegeben sei eine Grammatik wie folgt (ε ist Leersymbol):

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

Startsymbol = *Start*

Terminale = {*a, b, c, d*}

Nicht-Terminale = {*X, Y* }

Regeln = *{Start* $\rightarrow$ *X,*
Start $\rightarrow$ *Y,*
Start $\rightarrow$ *XY,*
X $\rightarrow$ *ab,*
X $\rightarrow$ *abX,*
Y $\rightarrow$ *cd,*
Y $\rightarrow$ *cYd* }

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

- ε ja / nein
- abba ja / nein
- acdc ja / nein
- abcd ja / nein
- aabccd ja / nein
- aabbccdd ja / nein
- ababccdd ja / nein
- abab ja / nein
- cd ja / nein
- ababacd ja / nein

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

Name:_____ Matr.-Nr: _____

Aufgabe 3: ADT Stack

Implementieren Sie einen Stack in Pseudocode auf Basis eines Arrays mit den Funktionen:

- size() gibt die Anzahl der Elemente im Stack zurück
- isEmpty() liefert **true**, wenn der Stack leer ist, **false** sonst
- top() liefert das oberste Element des Stacks, ohne es zu entfernen, **false**, falls der Stack leer ist
- push(Obj) fügt das Element **Obj** oben in den Stack ein, **true**, falls erfolgreich, **false**, falls der Stack voll ist
- pop() liefert das oberste Element des Stacks und entfernt es aus dem Stack, **false**, falls der Stack leer ist.

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:

- [Esther, koenig] = [roland, Seiffert].

- esther(Koenig) = roland(Seiffert).

- member(3, [2, 4, 6]) = member(X, [2, 4, 6])

- member(3, L), L = [1, 2, 3, 4].

Name:_____ Matr.-Nr: _____

Aufgabe 5: Prolog Programmanalyse

Gegeben ein Prolog-Programm foo(List1, Elem1, List2, Elem2) wie folgt

```
foo(List,Elem1,List,Elem2) :-
        not(member(Elem1,List)), not(member(Elem2,List)).
foo(List1,Elem1,List2,Elem2) :-
        append(First,[Elem1|Rest],List1),
        append(First,[Elem2|Rest],List2).
```

Beschreiben Sie kurz, was das Programm macht:

Zu was unifiziert List in den folgende Beispielen:

```
foo([1, 2, 3, 4], 3, List, kai).
```

List =

```
foo(List, kai, [1, 2, 3, 4], 3).
```

List =

```
foo([1,2,3,4], 5, List, kai).
```

List =

Name:_____ Matr.-Nr: _____