

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

Note: \_\_\_\_\_

**1. Unifizieren die beiden Terme? Falls ja, wie lautet die Belegung der Variablen X,Y,Z, nach der Unifikation:**

a)  $g([X|Y], X) = g([1, 2, 3], 1)$ .

Unifiziert Ja/Nein, falls ja: X=\_\_\_\_\_ Y= \_\_\_\_\_

falls nein: warum nicht:\_\_\_\_\_

b)  $g([[X]|Y], Z) = g([1, 2, 3], 1)$ .

Unifiziert Ja/Nein, falls ja: X=\_\_\_\_\_ Y= \_\_\_\_\_ Z= \_\_\_\_\_

falls nein: warum nicht:\_\_\_\_\_

c)  $g([X|Y], Z) = g([1, 2, 3], Z)$ .

Unifiziert Ja/Nein, falls ja: X=\_\_\_\_\_ Y= \_\_\_\_\_ Z= \_\_\_\_\_

falls nein: warum nicht:\_\_\_\_\_

d)  $g([X|Y], Z) = g([1], Z)$ .

Unifiziert Ja/Nein, falls ja: X=\_\_\_\_\_ Y= \_\_\_\_\_ Z= \_\_\_\_\_

falls nein: warum nicht:\_\_\_\_\_

e)  $g([X|Y], Z) = k([1], Z)$ .

Unifiziert Ja/Nein, falls ja: X=\_\_\_\_\_ Y= \_\_\_\_\_ Z= \_\_\_\_\_

falls nein: warum nicht:\_\_\_\_\_

g)  $g([X|[Y]], Z) = g([1, 2], 3)$ .

Unifiziert Ja/Nein, falls ja: X=\_\_\_\_\_ Y= \_\_\_\_\_ Z= \_\_\_\_\_

falls nein: warum nicht:\_\_\_\_\_

h)  $g([1, 2, 3], X) = g(X, X)$ .

Unifiziert Ja/Nein, falls ja: X=\_\_\_\_\_

falls nein: warum nicht:\_\_\_\_\_

i)  $g([1, 2, 3], [1|1, 2, 3]) = g(X, [1|X])$ .

Unifiziert Ja/Nein, falls ja: X=\_\_\_\_\_

falls nein: warum nicht:\_\_\_\_\_

j)  $g([1, 2, 3], [1|[1, 2, 3]]) = g(X, [1|Y])$ .

Unifiziert Ja/Nein, falls ja: X=\_\_\_\_\_ Y= \_\_\_\_\_

falls nein: warum nicht:\_\_\_\_\_

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

**2. Gegeben ist folgendes Prologprogramm, das gegebene Verbindungen zwischen zwei Punkten definiert.**

```
verbindung(a,b).  
verbindung(b,c).  
verbindung(d,e).  
verbindung(e,f).  
verbindung(c,d).
```

**a) schreiben Sie ein Prädikat `zweite_station(X,Y)` das beweist, ob Y die übernächste Station von X ist. (im obigen Programm ist c die übernächste Station von a)**

**b) schreiben Sie ein Prädikat `weg(X,Y)` das beweist, ob es eine Verbindung zwischen X und Y gibt. Es muss keine direkte Verbindung sein.**

**c) schreiben Sie ein Prädikat `weg_laenge(X,Y,N)` das beweist, ob es eine Verbindung zwischen X und Y gibt und zählt, wie viele Stationen zwischen X und Y liegen.**

**d. Was passiert mit Ihren Prädikaten, die sie in b) bis c) definiert haben, wenn folgende neue Verbindung eingeführt wird.**

```
verbindung(f,a).
```

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

**3. Gegeben sind folgende Prologregeln?**

```
wer_bin_ich(X, [X|L], L).           %Regel1
wer_bin_ich(X, [Y|L], [Y|L2]):-    %Regel2
  wer_bin_ich(X, L, L2).
```

**a) was ist das Ergebnis von L , wenn das Prädikat**

```
wer_bin_ich(1, [1,2,3],L).
```

**aufgerufen wird, wie nennen Sie dann das Prädikat.**

**b) Welche Schritte durchläuft das Prologprogramm beim Aufruf des Prädikats**

**wer\_bin\_ich ? Im folgenden sehen Sie eine Ausgabe ähnlich wie beim Tracer. Bitte tragen sie, falls vorhanden, die Werte der Variablen X, Y, L und L2 ein, die sie beim jeweiligen Aufruf annehmen:**

```
wer_bin_ich(3, [1,2,3,4],L).
```

```
CALL: wer_bin_ich( ... ) X=_____ Y=_____ L=_____ L2=_____??
```

```
CALL: wer_bin_ich( ... ) X=_____ Y=_____ L=_____ L2=_____??
```

```
CALL: wer_bin_ich( ... ) X=_____ Y=_____ L=_____ L2=_____??
```

```
eventuell: CALL: wer_bin_ich( ... ) X=_____ Y=_____ L=_____ L2=_____
```

```
eventuell: CALL: wer_bin_ich( ... ) X=_____ Y=_____ L=_____ L2=_____
```

```
L= _____
```

**c) was ist das Ergebnis von L , wenn das Prädikat**

```
wer_bin_ich(1, [1,2,3,1],L).
```

**aufgerufen wird.**

**d) Verändern Sie das Programm, damit die Aktion, die das Prädikat wer\_bin\_ich mit dem Element X durchführt, auf alle Elemente in L, die den selben Wert wie X haben angewandt wird.**

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

**4. Gegeben ist folgende DCG:**

```
s --> np, vp.  
np --> det, n.  
vp --> v, np.  
vp --> v.  
det --> [der].  
n --> [ball].  
n --> [mann].  
v --> [hüpft].
```

**a) Wie wird die in Prolog die oben definierte DCG aufgerufen um zu überprüfen, ob der folgende Eingabesatz die Regeln der DCG erfüllt:**

```
der ball hüpft
```

**b) Wie lautet die äquivalente Darstellung der obigen DCG mit Prologregeln, Argumenten und append.**

**c) Wie muss die DCG geändert werden, damit auch folgende Sätze erzeugt werden können? Was ist hier das Problem?**

```
der ball hüpft und der mann hüpft.  
der mann hüpft und der ball hüpft.
```

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

**5. Schreiben Sie ein Prädikat `member(E, L)` , das überprüft, ob ein Element **E** in der Liste **L** vorkommt.**

Beispielaufruf: `member(3, [1, 2, 3])`.