

Aufgabe 01 (12 Punkte)

Schreiben Sie ein Prädikat `geradeAnzUng(L)`, das prüft, ob eine Liste `L` mit positiven Zahlen eine gerade Anzahl von ungeraden Zahlen enthält. Die Prädikate `gerade(N)` respektive `ungerade(N)` seien gegeben und liefern genau dann `true` falls `N` eine gerade respektive ungerade Zahl ist.

Es reicht, wenn das Prädikat in einer Richtung funktioniert.

```
geradeAnzUng([]):-  
    !.  
geradeAnzUng(L):-  
    count(L,0,Z),  
    gerade(Z).  
  
count([],A,A):-  
    !.  
count([H|T],A,E):-  
    ungerade(H),  
    !,  
    NA is A +1,  
    count(T,NA,E).  
  
count([_|T],A,E):-  
    count(T,A,E).
```

Alternative:

```
zaehleungerade([],0). % Zählt die ungeraden Zahlen in der Liste
```

```
zaehleungerade([H|T],N):- !.  
    gerade(H),!,  
    zaehleungerade(T,N)  
  
    zaehleungerade([H|T],N):-  
    ungerade(H),  
    zaehleungerade(T,N1),  
    N is N1+1.  
  
geradeAnzUngerade(L) :-  
    zaehleungerade(L,N),  
    gerade(N).
```

Aufgabe 2 (12 Punkte)

- Definieren Sie ein Prädikat `geschwister(A, B)`, das genau dann erfüllt ist, wenn A und B Geschwister voneinander sind.
- Definieren Sie ein Prädikat `nichte(N, ToO)`, das genau dann wahr ist, wenn ToO Tante oder Onkel der Nichte N ist. (Eine Nichte N von ToO ist das weibliche Kind eines Bruders oder einer Schwester von ToO).

```
maennlich(anton).
maennlich(karl).
maennlich(hans).
maennlich(manfred).
maennlich(paul).
maennlich(otto).
maennlich(klaus).
Maennlich(willi).
weiblich(gerda).
weiblich(hilde).
weiblich(ida).
weiblich(inge).
weiblich(maria).
familie(anton, gerda, [karl, hans, ida]).
familie(karl, hilde, [manfred, willi]).
familie(hans, maria, [inge, paul]).
familie(otto, ida, [dieter, klaus]).
```

```
geschwister(A, B) :-
    Familie(_, _, K),
    member(A, K),
    member(B, K),
    A \= B.
```

```
nichte(N, Too) :-
    geschwister(Too, G),
    (familie(G, _, K1); familie(_G, K1)),
    member(N, K1),
    weiblich(N).
```

Aufgabe 3 (12 Punkte)

- a) Schreiben Sie ein Prolog-Prädikat `zip(L1, L2, L3)`, das in der Liste `L3`, die Elemente der Listen `L1` und `L2` abwechselnd aufsammelt (Reißverschlussverfahren), beginnend mit dem 1. Element von `L1`. Ist eine der beiden Listen `L1` oder `L2` erschöpft, wird der Rest der anderen an `L3` angefügt. Beispiel `zip([1,2,3,4], [a,b], [1,a,2,b,3,4])` liefert `true`.
- b) Schreiben Sie ein Prädikat `split(L, U, G)`, das die Liste `L` so zerlegt: in `U` sind die Elemente von `L`, die in `L` an ungerader Position stehen, in `G` die an gerader Position. Beispiel: `split([1,2,3], [1,3], [2])` liefert `true`.

```
zip(L1, [], L1):- !.  
zip([], L2, L2):- !.  
zip([H1|T1], [H2|T2], [H1,H2|T3]) :-  
    zip(T1,T2,T3).
```

```
split([], [], []).  
split([X,Y|R], [X|R1], [Y|R2]) :- split(R,R1,R2).  
split([X], [X], []).
```

Aufgabe 04 (12 Punkte)

Gegeben sei die Klauselmeng e $S = \{\{\neg A, \neg B, \neg C\}, \{B, \neg C\}, \{\neg A, C\}, \{A, \neg B\}, \{A, C\}\}$

Zeigen Sie mit Hilfe des Resolutionskalküls, dass S unerfüllbar ist.

