1.3

`assert` = `assertz`: inserts at the end of the clause base.

`asserta` : inserts at beginning.

a)

```
mutter_von( charlotte , barbara ).
true;
```

b)

```
?- vater_von(walter, andrea).
false.
```

c)

```
?- mutter_von( X , andrea ).
X = barbara.
```

d)

```
?- mutter_von( X , johannes ).
false.
```

e)

```
?- mutter_von( charlotte, X ).
X = barbara ;
X = magdalena.
```

f)

```
?- listing(mutter_von).
:- dynamic mutter_von/2.

mutter_von(marie, hans).
mutter_von(marie, helga).
mutter_von(julia, otto).
mutter_von(barbara, klaus).
mutter_von(barbara, andrea).
mutter_von(charlotte, barbara).
mutter_von(charlotte, magdalena).
```

g)

```
?- not(mutter_von(helga, X)).
true.
```

h)

```
?- not(mutter_von(barbara, X)).
false.
```

i)

```
?- not(not(mutter_von(barbara, X))).
true.
```

2.

Füge folgendes zu `familie.pl` hinzu:

```
elternteil_von(A,B) :- vater_von(A,B) ; mutter_von(A,B).
enkel_von(C,A) :- elternteil_von(A,B), elternteil_von(B, C).
```

Dann stelle folgende Abfrage:

```
?- enkel_von(X, charlotte).
X = klaus ;
X = andrea ;
```