

```

vorfahr(V,N) :- elternteil(V,N).
vorfahr(V,N) :- elternteil(Z,N), vorfahr(V,Z).

```

(Regel 1)
(Regel 2)

```

vorfahr( V , frank).
Regel 1:      elternteil( V , frank).
Ausgabe:     V = ares.
Regel 2:      elternteil( Z , frank), vorfahr( V , Z ).

```

```

R: elternteil( Z , frank).
A: Z = ares.
UND
R: vorfahr( V , ares).

```

Rekursion:

```

vorfahr( V , ares).
Regel 1:      elternteil( V , _____).
Ausgabe:     V = _____:
              V = _____:
Regel 2:      elternteil( Z , _____), vorfahr( V , Z ).

```

```

R: elternteil( Z , _____).
A: Z = _____:
    Z = _____:
UND
R: vorfahr( V , _____).
   vorfahr( V , _____).

```

Rekursion:

```

vorfahr( V , zeus).
Regel 1:      elternteil( V , _____).
Ausgabe:     false.
Regel 2:      elternteil( Z , _____), vorfahr( V , Z ).

```

```

R: elternteil( Z , _____).
A: false. [Abbruch]

```

Rekursion:

```

vorfahr( V , hera).
Regel 1:      elternteil( V , _____).
Ausgabe:     false.
Regel 2:      elternteil( Z , _____), vorfahr( V,Z ).

```

```

R: elternteil( Z , _____).
A: false. [Abbruch]

```