**Exercise 1: The Smart Phone Rivalry**

SumSum, a competitor of Appy, developed some nice smart phone technology called

Galactica-S3, all of which was stolen by Stevey, who is a Boss. It is unethical for a Boss

to steal business from rival companies. A competitor of Appy is a rival. Smart phone

technology is a business.

1. **Translate the natural language statements above describing the dealing within the Smart Phone industry in to First Order Logic, (FOL).**

Competitor(SumSum, Appy)

∃x Smartphone(x) ^ Developed(SumSum, x)

∀x Smartphone(x) ^ Developed(SumSum, x) => Steal(Stevey, x, SumSum)

Boss(Stevey)

∀x, y, z Boss(x) ^ Business(y) ^ Company(z) ^ Rival(z) ^ Steal(x, y, z) => unethical(x)

∀x competitor(x, Appy) => Rival(x)

∀x Smartphone(x) => Business(x)

Company(SumSum)

Company(Appy)

1. **Write these FOL statements as Prolog clauses.**

company(sumSum).

company(appy).

boss(stevey).

technology(galacticaS3).

competitor(sumSum, appy).

develop(sumSum, galacticaS3).

steal(stevey, galacticaS3).

rival(X):- competitor(X, appy); competitor(appy, X).

business(X):- technology(X).

unethical(X):- boss(X), rival(Comp), business(Biz), develop(Comp, Tech), steal(X, Biz).

1. **Using the prolog search engine, prove that Stevey is unethical. Show a traceof your proof.**

[trace] 2 ?- unethical(stevey).

Call: (8) unethical(stevey) ? creep

Call: (9) boss(stevey) ? creep

Exit: (9) boss(stevey) ? creep

Call: (9) rival(\_3982) ? creep

Call: (10) competitor(\_3982, appy) ? creep

Exit: (10) competitor(sumSum, appy) ? creep

Exit: (9) rival(sumSum) ? creep

Call: (9) business(\_3982) ? creep

Call: (10) technology(\_3982) ? creep

Exit: (10) technology(galacticaS3) ? creep

Exit: (9) business(galacticaS3) ? creep

Call: (9) develop(sumSum, \_3984) ? creep

Exit: (9) develop(sumSum, galacticaS3) ? creep

Call: (9) steal(stevey, galacticaS3) ? creep

Exit: (9) steal(stevey, galacticaS3) ? creep

Exit: (8) unethical(stevey) ? creep

true .

**Exercise 2: The Royal Family**

The old Royal succession rule states that the throne is passed down along the male line according to the order of birth before the consideration along the female line – similarly according to the order of birth. Queen Elizabeth, the monarch of United Kingdom, has four offsprings; namely:- Prince Charles, Princess Ann, Prince Andrew and Prince Edward – listed in the order of birth.

1. **Define their relations and rules in a prolog rule base. Hence, define the old Royal succession rule. Using this old succession rule determine the line of succession based on the information given. Do a trace to show your results.**

offspring(prince, charlie).

offspring(princess, ann).

offspring(prince, andrew).

offspring(prince, edward).

elder(charlie, ann).

elder(ann, andrew).

elder(andrew, edward).

is\_elder(A, B):- elder(A, B).

is\_elder(A, B):- elder(A, C), is\_elder(C, B).

male(A):- offspring(prince, A).

female(A):- offspring(princess, A).

is\_youngest\_male(A):- is\_elder(B, A), is\_elder(C, B), male(A), male(B), male(C).

is\_higher(A, B):- male(A), female(B).

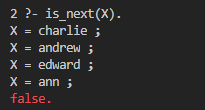
is\_higher(A, B):- (male(A), male(B)) ; (female(A), female(B)), is\_elder(A, B).

is\_next(A):-

(is\_higher(A, B), female(B));

(is\_higher(A, B), female(A), female(B));

(is\_higher(B, A), female(A), is\_youngest\_male(B)).



**Trace**

[trace] 5 ?- is\_next(X).

Call: (8) is\_next(\_7106) ? creep

Call: (9) is\_higher(\_7106, \_7326) ? creep

Call: (10) male(\_7106) ? creep

Call: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Call: (10) female(\_7324) ? creep

Call: (11) offspring(princess, \_7326) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Exit: (9) is\_higher(charlie, ann) ? creep

Call: (9) female(ann) ? creep

Call: (10) offspring(princess, ann) ? creep

Exit: (10) offspring(princess, ann) ? creep

Exit: (9) female(ann) ? creep

Exit: (8) is\_next(charlie) ? creep

X = charlie ;

Redo: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Call: (10) female(\_7324) ? creep

Call: (11) offspring(princess, \_7326) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Exit: (9) is\_higher(andrew, ann) ? creep

Call: (9) female(ann) ? creep

Call: (10) offspring(princess, ann) ? creep

Exit: (10) offspring(princess, ann) ? creep

Exit: (9) female(ann) ? creep

Exit: (8) is\_next(andrew) ? creep

X = andrew ;

Redo: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Call: (10) female(\_7324) ? creep

Call: (11) offspring(princess, \_7326) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Exit: (9) is\_higher(edward, ann) ? creep

Call: (9) female(ann) ? creep

Call: (10) offspring(princess, ann) ? creep

Exit: (10) offspring(princess, ann) ? creep

Exit: (9) female(ann) ? creep

Exit: (8) is\_next(edward) ? creep

X = edward ;

Redo: (9) is\_higher(\_7106, \_7326) ? creep

Call: (10) male(\_7106) ? creep

Call: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Call: (10) male(\_7324) ? creep

Call: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Exit: (9) is\_higher(charlie, charlie) ? creep

Call: (9) female(charlie) ? creep

Call: (10) offspring(princess, charlie) ? creep

Fail: (10) offspring(princess, charlie) ? creep

Fail: (9) female(charlie) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Exit: (9) is\_higher(charlie, andrew) ? creep

Call: (9) female(andrew) ? creep

Call: (10) offspring(princess, andrew) ? creep

Fail: (10) offspring(princess, andrew) ? creep

Fail: (9) female(andrew) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Exit: (9) is\_higher(charlie, edward) ? creep

Call: (9) female(edward) ? creep

Call: (10) offspring(princess, edward) ? creep

Fail: (10) offspring(princess, edward) ? creep

Fail: (9) female(edward) ? creep

Redo: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Call: (10) male(\_7324) ? creep

Call: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Exit: (9) is\_higher(andrew, charlie) ? creep

Call: (9) female(charlie) ? creep

Call: (10) offspring(princess, charlie) ? creep

Fail: (10) offspring(princess, charlie) ? creep

Fail: (9) female(charlie) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Exit: (9) is\_higher(andrew, andrew) ? creep

Call: (9) female(andrew) ? creep

Call: (10) offspring(princess, andrew) ? creep

Fail: (10) offspring(princess, andrew) ? creep

Fail: (9) female(andrew) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Exit: (9) is\_higher(andrew, edward) ? creep

Call: (9) female(edward) ? creep

Call: (10) offspring(princess, edward) ? creep

Fail: (10) offspring(princess, edward) ? creep

Fail: (9) female(edward) ? creep

Redo: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Call: (10) male(\_7324) ? creep

Call: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Exit: (9) is\_higher(edward, charlie) ? creep

Call: (9) female(charlie) ? creep

Call: (10) offspring(princess, charlie) ? creep

Fail: (10) offspring(princess, charlie) ? creep

Fail: (9) female(charlie) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Exit: (9) is\_higher(edward, andrew) ? creep

Call: (9) female(andrew) ? creep

Call: (10) offspring(princess, andrew) ? creep

Fail: (10) offspring(princess, andrew) ? creep

Fail: (9) female(andrew) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Exit: (9) is\_higher(edward, edward) ? creep

Call: (9) female(edward) ? creep

Call: (10) offspring(princess, edward) ? creep

Fail: (10) offspring(princess, edward) ? creep

Fail: (9) female(edward) ? creep

Redo: (9) is\_higher(\_7106, \_7326) ? creep

Call: (10) female(\_7106) ? creep

Call: (11) offspring(princess, \_7106) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Call: (10) female(\_7324) ? creep

Call: (11) offspring(princess, \_7326) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Call: (10) is\_elder(ann, ann) ? creep

Call: (11) elder(ann, ann) ? creep

Fail: (11) elder(ann, ann) ? creep

Redo: (10) is\_elder(ann, ann) ? creep

Call: (11) elder(ann, \_7326) ? creep

Exit: (11) elder(ann, andrew) ? creep

Call: (11) is\_elder(andrew, ann) ? creep

Call: (12) elder(andrew, ann) ? creep

Fail: (12) elder(andrew, ann) ? creep

Redo: (11) is\_elder(andrew, ann) ? creep

Call: (12) elder(andrew, \_7326) ? creep

Exit: (12) elder(andrew, edward) ? creep

Call: (12) is\_elder(edward, ann) ? creep

Call: (13) elder(edward, ann) ? creep

Fail: (13) elder(edward, ann) ? creep

Redo: (12) is\_elder(edward, ann) ? creep

Call: (13) elder(edward, \_7326) ? creep

Fail: (13) elder(edward, \_7326) ? creep

Fail: (12) is\_elder(edward, ann) ? creep

Fail: (11) is\_elder(andrew, ann) ? creep

Fail: (10) is\_elder(ann, ann) ? creep

Fail: (9) is\_higher(\_7106, \_7326) ? creep

Redo: (8) is\_next(\_7106) ? creep

Call: (9) is\_higher(\_7106, \_7326) ? creep

Call: (10) male(\_7106) ? creep

Call: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Call: (10) female(\_7324) ? creep

Call: (11) offspring(princess, \_7326) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Exit: (9) is\_higher(charlie, ann) ? creep

Call: (9) female(charlie) ? creep

Call: (10) offspring(princess, charlie) ? creep

Fail: (10) offspring(princess, charlie) ? creep

Fail: (9) female(charlie) ? creep

Redo: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Call: (10) female(\_7324) ? creep

Call: (11) offspring(princess, \_7326) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Exit: (9) is\_higher(andrew, ann) ? creep

Call: (9) female(andrew) ? creep

Call: (10) offspring(princess, andrew) ? creep

Fail: (10) offspring(princess, andrew) ? creep

Fail: (9) female(andrew) ? creep

Redo: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Call: (10) female(\_7324) ? creep

Call: (11) offspring(princess, \_7326) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Exit: (9) is\_higher(edward, ann) ? creep

Call: (9) female(edward) ? creep

Call: (10) offspring(princess, edward) ? creep

Fail: (10) offspring(princess, edward) ? creep

Fail: (9) female(edward) ? creep

Redo: (9) is\_higher(\_7106, \_7326) ? creep

Call: (10) male(\_7106) ? creep

Call: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Call: (10) male(\_7324) ? creep

Call: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Exit: (9) is\_higher(charlie, charlie) ? creep

Call: (9) female(charlie) ? creep

Call: (10) offspring(princess, charlie) ? creep

Fail: (10) offspring(princess, charlie) ? creep

Fail: (9) female(charlie) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Exit: (9) is\_higher(charlie, andrew) ? creep

Call: (9) female(charlie) ? creep

Call: (10) offspring(princess, charlie) ? creep

Fail: (10) offspring(princess, charlie) ? creep

Fail: (9) female(charlie) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Exit: (9) is\_higher(charlie, edward) ? creep

Call: (9) female(charlie) ? creep

Call: (10) offspring(princess, charlie) ? creep

Fail: (10) offspring(princess, charlie) ? creep

Fail: (9) female(charlie) ? creep

Redo: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Call: (10) male(\_7324) ? creep

Call: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Exit: (9) is\_higher(andrew, charlie) ? creep

Call: (9) female(andrew) ? creep

Call: (10) offspring(princess, andrew) ? creep

Fail: (10) offspring(princess, andrew) ? creep

Fail: (9) female(andrew) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Exit: (9) is\_higher(andrew, andrew) ? creep

Call: (9) female(andrew) ? creep

Call: (10) offspring(princess, andrew) ? creep

Fail: (10) offspring(princess, andrew) ? creep

Fail: (9) female(andrew) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Exit: (9) is\_higher(andrew, edward) ? creep

Call: (9) female(andrew) ? creep

Call: (10) offspring(princess, andrew) ? creep

Fail: (10) offspring(princess, andrew) ? creep

Fail: (9) female(andrew) ? creep

Redo: (11) offspring(prince, \_7106) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Call: (10) male(\_7324) ? creep

Call: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Exit: (9) is\_higher(edward, charlie) ? creep

Call: (9) female(edward) ? creep

Call: (10) offspring(princess, edward) ? creep

Fail: (10) offspring(princess, edward) ? creep

Fail: (9) female(edward) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Exit: (9) is\_higher(edward, andrew) ? creep

Call: (9) female(edward) ? creep

Call: (10) offspring(princess, edward) ? creep

Fail: (10) offspring(princess, edward) ? creep

Fail: (9) female(edward) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Exit: (9) is\_higher(edward, edward) ? creep

Call: (9) female(edward) ? creep

Call: (10) offspring(princess, edward) ? creep

Fail: (10) offspring(princess, edward) ? creep

Fail: (9) female(edward) ? creep

Redo: (9) is\_higher(\_7106, \_7326) ? creep

Call: (10) female(\_7106) ? creep

Call: (11) offspring(princess, \_7106) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Call: (10) female(\_7324) ? creep

Call: (11) offspring(princess, \_7326) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Call: (10) is\_elder(ann, ann) ? creep

Call: (11) elder(ann, ann) ? creep

Fail: (11) elder(ann, ann) ? creep

Redo: (10) is\_elder(ann, ann) ? creep

Call: (11) elder(ann, \_7326) ? creep

Exit: (11) elder(ann, andrew) ? creep

Call: (11) is\_elder(andrew, ann) ? creep

Call: (12) elder(andrew, ann) ? creep

Fail: (12) elder(andrew, ann) ? creep

Redo: (11) is\_elder(andrew, ann) ? creep

Call: (12) elder(andrew, \_7326) ? creep

Exit: (12) elder(andrew, edward) ? creep

Call: (12) is\_elder(edward, ann) ? creep

Call: (13) elder(edward, ann) ? creep

Fail: (13) elder(edward, ann) ? creep

Redo: (12) is\_elder(edward, ann) ? creep

Call: (13) elder(edward, \_7326) ? creep

Fail: (13) elder(edward, \_7326) ? creep

Fail: (12) is\_elder(edward, ann) ? creep

Fail: (11) is\_elder(andrew, ann) ? creep

Fail: (10) is\_elder(ann, ann) ? creep

Fail: (9) is\_higher(\_7106, \_7326) ? creep

Redo: (8) is\_next(\_7106) ? creep

Call: (9) is\_higher(\_7324, \_7106) ? creep

Call: (10) male(\_7324) ? creep

Call: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Call: (10) female(\_7106) ? creep

Call: (11) offspring(princess, \_7106) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Exit: (9) is\_higher(charlie, ann) ? creep

Call: (9) female(ann) ? creep

Call: (10) offspring(princess, ann) ? creep

Exit: (10) offspring(princess, ann) ? creep

Exit: (9) female(ann) ? creep

Call: (9) is\_youngest\_male(charlie) ? creep

Call: (10) is\_elder(\_7324, charlie) ? creep

Call: (11) elder(\_7324, charlie) ? creep

Fail: (11) elder(\_7324, charlie) ? creep

Redo: (10) is\_elder(\_7324, charlie) ? creep

Call: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(charlie, ann) ? creep

Call: (11) is\_elder(ann, charlie) ? creep

Call: (12) elder(ann, charlie) ? creep

Fail: (12) elder(ann, charlie) ? creep

Redo: (11) is\_elder(ann, charlie) ? creep

Call: (12) elder(ann, \_7326) ? creep

Exit: (12) elder(ann, andrew) ? creep

Call: (12) is\_elder(andrew, charlie) ? creep

Call: (13) elder(andrew, charlie) ? creep

Fail: (13) elder(andrew, charlie) ? creep

Redo: (12) is\_elder(andrew, charlie) ? creep

Call: (13) elder(andrew, \_7326) ? creep

Exit: (13) elder(andrew, edward) ? creep

Call: (13) is\_elder(edward, charlie) ? creep

Call: (14) elder(edward, charlie) ? creep

Fail: (14) elder(edward, charlie) ? creep

Redo: (13) is\_elder(edward, charlie) ? creep

Call: (14) elder(edward, \_7326) ? creep

Fail: (14) elder(edward, \_7326) ? creep

Fail: (13) is\_elder(edward, charlie) ? creep

Fail: (12) is\_elder(andrew, charlie) ? creep

Fail: (11) is\_elder(ann, charlie) ? creep

Redo: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(ann, andrew) ? creep

Call: (11) is\_elder(andrew, charlie) ? creep

Call: (12) elder(andrew, charlie) ? creep

Fail: (12) elder(andrew, charlie) ? creep

Redo: (11) is\_elder(andrew, charlie) ? creep

Call: (12) elder(andrew, \_7326) ? creep

Exit: (12) elder(andrew, edward) ? creep

Call: (12) is\_elder(edward, charlie) ? creep

Call: (13) elder(edward, charlie) ? creep

Fail: (13) elder(edward, charlie) ? creep

Redo: (12) is\_elder(edward, charlie) ? creep

Call: (13) elder(edward, \_7326) ? creep

Fail: (13) elder(edward, \_7326) ? creep

Fail: (12) is\_elder(edward, charlie) ? creep

Fail: (11) is\_elder(andrew, charlie) ? creep

Redo: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(andrew, edward) ? creep

Call: (11) is\_elder(edward, charlie) ? creep

Call: (12) elder(edward, charlie) ? creep

Fail: (12) elder(edward, charlie) ? creep

Redo: (11) is\_elder(edward, charlie) ? creep

Call: (12) elder(edward, \_7326) ? creep

Fail: (12) elder(edward, \_7326) ? creep

Fail: (11) is\_elder(edward, charlie) ? creep

Fail: (10) is\_elder(\_7324, charlie) ? creep

Fail: (9) is\_youngest\_male(charlie) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Call: (10) female(\_7106) ? creep

Call: (11) offspring(princess, \_7106) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Exit: (9) is\_higher(andrew, ann) ? creep

Call: (9) female(ann) ? creep

Call: (10) offspring(princess, ann) ? creep

Exit: (10) offspring(princess, ann) ? creep

Exit: (9) female(ann) ? creep

Call: (9) is\_youngest\_male(andrew) ? creep

Call: (10) is\_elder(\_7324, andrew) ? creep

Call: (11) elder(\_7324, andrew) ? creep

Exit: (11) elder(ann, andrew) ? creep

Exit: (10) is\_elder(ann, andrew) ? creep

Call: (10) is\_elder(\_7324, ann) ? creep

Call: (11) elder(\_7324, ann) ? creep

Exit: (11) elder(charlie, ann) ? creep

Exit: (10) is\_elder(charlie, ann) ? creep

Call: (10) male(andrew) ? creep

Call: (11) offspring(prince, andrew) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Call: (10) male(ann) ? creep

Call: (11) offspring(prince, ann) ? creep

Fail: (11) offspring(prince, ann) ? creep

Fail: (10) male(ann) ? creep

Redo: (11) offspring(prince, andrew) ? creep

Fail: (11) offspring(prince, andrew) ? creep

Fail: (10) male(andrew) ? creep

Redo: (10) is\_elder(\_7324, ann) ? creep

Call: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(charlie, ann) ? creep

Call: (11) is\_elder(ann, ann) ? creep

Call: (12) elder(ann, ann) ? creep

Fail: (12) elder(ann, ann) ? creep

Redo: (11) is\_elder(ann, ann) ? creep

Call: (12) elder(ann, \_7326) ? creep

Exit: (12) elder(ann, andrew) ? creep

Call: (12) is\_elder(andrew, ann) ? creep

Call: (13) elder(andrew, ann) ? creep

Fail: (13) elder(andrew, ann) ? creep

Redo: (12) is\_elder(andrew, ann) ? creep

Call: (13) elder(andrew, \_7326) ? creep

Exit: (13) elder(andrew, edward) ? creep

Call: (13) is\_elder(edward, ann) ? creep

Call: (14) elder(edward, ann) ? creep

Fail: (14) elder(edward, ann) ? creep

Redo: (13) is\_elder(edward, ann) ? creep

Call: (14) elder(edward, \_7326) ? creep

Fail: (14) elder(edward, \_7326) ? creep

Fail: (13) is\_elder(edward, ann) ? creep

Fail: (12) is\_elder(andrew, ann) ? creep

Fail: (11) is\_elder(ann, ann) ? creep

Redo: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(ann, andrew) ? creep

Call: (11) is\_elder(andrew, ann) ? creep

Call: (12) elder(andrew, ann) ? creep

Fail: (12) elder(andrew, ann) ? creep

Redo: (11) is\_elder(andrew, ann) ? creep

Call: (12) elder(andrew, \_7326) ? creep

Exit: (12) elder(andrew, edward) ? creep

Call: (12) is\_elder(edward, ann) ? creep

Call: (13) elder(edward, ann) ? creep

Fail: (13) elder(edward, ann) ? creep

Redo: (12) is\_elder(edward, ann) ? creep

Call: (13) elder(edward, \_7326) ? creep

Fail: (13) elder(edward, \_7326) ? creep

Fail: (12) is\_elder(edward, ann) ? creep

Fail: (11) is\_elder(andrew, ann) ? creep

Redo: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(andrew, edward) ? creep

Call: (11) is\_elder(edward, ann) ? creep

Call: (12) elder(edward, ann) ? creep

Fail: (12) elder(edward, ann) ? creep

Redo: (11) is\_elder(edward, ann) ? creep

Call: (12) elder(edward, \_7326) ? creep

Fail: (12) elder(edward, \_7326) ? creep

Fail: (11) is\_elder(edward, ann) ? creep

Fail: (10) is\_elder(\_7324, ann) ? creep

Redo: (10) is\_elder(\_7324, andrew) ? creep

Call: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(charlie, ann) ? creep

Call: (11) is\_elder(ann, andrew) ? creep

Call: (12) elder(ann, andrew) ? creep

Exit: (12) elder(ann, andrew) ? creep

Exit: (11) is\_elder(ann, andrew) ? creep

Exit: (10) is\_elder(charlie, andrew) ? creep

Call: (10) is\_elder(\_7324, charlie) ? creep

Call: (11) elder(\_7324, charlie) ? creep

Fail: (11) elder(\_7324, charlie) ? creep

Redo: (10) is\_elder(\_7324, charlie) ? creep

Call: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(charlie, ann) ? creep

Call: (11) is\_elder(ann, charlie) ? creep

Call: (12) elder(ann, charlie) ? creep

Fail: (12) elder(ann, charlie) ? creep

Redo: (11) is\_elder(ann, charlie) ? creep

Call: (12) elder(ann, \_7326) ? creep

Exit: (12) elder(ann, andrew) ? creep

Call: (12) is\_elder(andrew, charlie) ? creep

Call: (13) elder(andrew, charlie) ? creep

Fail: (13) elder(andrew, charlie) ? creep

Redo: (12) is\_elder(andrew, charlie) ? creep

Call: (13) elder(andrew, \_7326) ? creep

Exit: (13) elder(andrew, edward) ? creep

Call: (13) is\_elder(edward, charlie) ? creep

Call: (14) elder(edward, charlie) ? creep

Fail: (14) elder(edward, charlie) ? creep

Redo: (13) is\_elder(edward, charlie) ? creep

Call: (14) elder(edward, \_7326) ? creep

Fail: (14) elder(edward, \_7326) ? creep

Fail: (13) is\_elder(edward, charlie) ? creep

Fail: (12) is\_elder(andrew, charlie) ? creep

Fail: (11) is\_elder(ann, charlie) ? creep

Redo: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(ann, andrew) ? creep

Call: (11) is\_elder(andrew, charlie) ? creep

Call: (12) elder(andrew, charlie) ? creep

Fail: (12) elder(andrew, charlie) ? creep

Redo: (11) is\_elder(andrew, charlie) ? creep

Call: (12) elder(andrew, \_7326) ? creep

Exit: (12) elder(andrew, edward) ? creep

Call: (12) is\_elder(edward, charlie) ? creep

Call: (13) elder(edward, charlie) ? creep

Fail: (13) elder(edward, charlie) ? creep

Redo: (12) is\_elder(edward, charlie) ? creep

Call: (13) elder(edward, \_7326) ? creep

Fail: (13) elder(edward, \_7326) ? creep

Fail: (12) is\_elder(edward, charlie) ? creep

Fail: (11) is\_elder(andrew, charlie) ? creep

Redo: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(andrew, edward) ? creep

Call: (11) is\_elder(edward, charlie) ? creep

Call: (12) elder(edward, charlie) ? creep

Fail: (12) elder(edward, charlie) ? creep

Redo: (11) is\_elder(edward, charlie) ? creep

Call: (12) elder(edward, \_7326) ? creep

Fail: (12) elder(edward, \_7326) ? creep

Fail: (11) is\_elder(edward, charlie) ? creep

Fail: (10) is\_elder(\_7324, charlie) ? creep

Redo: (11) is\_elder(ann, andrew) ? creep

Call: (12) elder(ann, \_7326) ? creep

Exit: (12) elder(ann, andrew) ? creep

Call: (12) is\_elder(andrew, andrew) ? creep

Call: (13) elder(andrew, andrew) ? creep

Fail: (13) elder(andrew, andrew) ? creep

Redo: (12) is\_elder(andrew, andrew) ? creep

Call: (13) elder(andrew, \_7326) ? creep

Exit: (13) elder(andrew, edward) ? creep

Call: (13) is\_elder(edward, andrew) ? creep

Call: (14) elder(edward, andrew) ? creep

Fail: (14) elder(edward, andrew) ? creep

Redo: (13) is\_elder(edward, andrew) ? creep

Call: (14) elder(edward, \_7326) ? creep

Fail: (14) elder(edward, \_7326) ? creep

Fail: (13) is\_elder(edward, andrew) ? creep

Fail: (12) is\_elder(andrew, andrew) ? creep

Fail: (11) is\_elder(ann, andrew) ? creep

Redo: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(ann, andrew) ? creep

Call: (11) is\_elder(andrew, andrew) ? creep

Call: (12) elder(andrew, andrew) ? creep

Fail: (12) elder(andrew, andrew) ? creep

Redo: (11) is\_elder(andrew, andrew) ? creep

Call: (12) elder(andrew, \_7326) ? creep

Exit: (12) elder(andrew, edward) ? creep

Call: (12) is\_elder(edward, andrew) ? creep

Call: (13) elder(edward, andrew) ? creep

Fail: (13) elder(edward, andrew) ? creep

Redo: (12) is\_elder(edward, andrew) ? creep

Call: (13) elder(edward, \_7326) ? creep

Fail: (13) elder(edward, \_7326) ? creep

Fail: (12) is\_elder(edward, andrew) ? creep

Fail: (11) is\_elder(andrew, andrew) ? creep

Redo: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(andrew, edward) ? creep

Call: (11) is\_elder(edward, andrew) ? creep

Call: (12) elder(edward, andrew) ? creep

Fail: (12) elder(edward, andrew) ? creep

Redo: (11) is\_elder(edward, andrew) ? creep

Call: (12) elder(edward, \_7326) ? creep

Fail: (12) elder(edward, \_7326) ? creep

Fail: (11) is\_elder(edward, andrew) ? creep

Fail: (10) is\_elder(\_7324, andrew) ? creep

Fail: (9) is\_youngest\_male(andrew) ? creep

Redo: (11) offspring(prince, \_7326) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Call: (10) female(\_7106) ? creep

Call: (11) offspring(princess, \_7106) ? creep

Exit: (11) offspring(princess, ann) ? creep

Exit: (10) female(ann) ? creep

Exit: (9) is\_higher(edward, ann) ? creep

Call: (9) female(ann) ? creep

Call: (10) offspring(princess, ann) ? creep

Exit: (10) offspring(princess, ann) ? creep

Exit: (9) female(ann) ? creep

Call: (9) is\_youngest\_male(edward) ? creep

Call: (10) is\_elder(\_7324, edward) ? creep

Call: (11) elder(\_7324, edward) ? creep

Exit: (11) elder(andrew, edward) ? creep

Exit: (10) is\_elder(andrew, edward) ? creep

Call: (10) is\_elder(\_7324, andrew) ? creep

Call: (11) elder(\_7324, andrew) ? creep

Exit: (11) elder(ann, andrew) ? creep

Exit: (10) is\_elder(ann, andrew) ? creep

Call: (10) male(edward) ? creep

Call: (11) offspring(prince, edward) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Call: (10) male(andrew) ? creep

Call: (11) offspring(prince, andrew) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Call: (10) male(ann) ? creep

Call: (11) offspring(prince, ann) ? creep

Fail: (11) offspring(prince, ann) ? creep

Fail: (10) male(ann) ? creep

Redo: (11) offspring(prince, andrew) ? creep

Fail: (11) offspring(prince, andrew) ? creep

Fail: (10) male(andrew) ? creep

Redo: (10) is\_elder(\_7324, andrew) ? creep

Call: (11) elder(\_7324, \_7326) ? creep

Exit: (11) elder(charlie, ann) ? creep

Call: (11) is\_elder(ann, andrew) ? creep

Call: (12) elder(ann, andrew) ? creep

Exit: (12) elder(ann, andrew) ? creep

Exit: (11) is\_elder(ann, andrew) ? creep

Exit: (10) is\_elder(charlie, andrew) ? creep

Call: (10) male(edward) ? creep

Call: (11) offspring(prince, edward) ? creep

Exit: (11) offspring(prince, edward) ? creep

Exit: (10) male(edward) ? creep

Call: (10) male(andrew) ? creep

Call: (11) offspring(prince, andrew) ? creep

Exit: (11) offspring(prince, andrew) ? creep

Exit: (10) male(andrew) ? creep

Call: (10) male(charlie) ? creep

Call: (11) offspring(prince, charlie) ? creep

Exit: (11) offspring(prince, charlie) ? creep

Exit: (10) male(charlie) ? creep

Exit: (9) is\_youngest\_male(edward) ? creep

Exit: (8) is\_next(ann) ? creep

X = ann .

1. **Recently, the Royal succession rule has been modified. The throne is now passed down according to the order of birth irrespective of gender. Modify your rules and prolog knowledge base to handle the new succession rule. Explain the necessary changes to the knowledge needed to represent the new information. Use this new succession rule to determine the new line of succession based on the same knowledge given. Show your results using a trace.**

offspring(prince, charlie).

offspring(princess, ann).

offspring(prince, andrew).

offspring(prince, edward).

elder(charlie, ann).

elder(ann, andrew).

elder(andrew, edward).

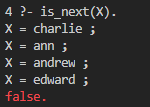
is\_elder(A, B):- elder(A, B).

is\_elder(A, B):- elder(A, C), is\_elder(C, B).

is\_youngest(A):- is\_elder(B, A), is\_elder(C, B), is\_elder(D, C).

is\_next(A):- elder(A, B).

is\_next(A):- (not(elder(A, B)), is\_next(B)) ; is\_youngest(A).



**Trace**

[trace] 5 ?- is\_next(X).

Call: (8) is\_next(\_5562) ? creep

Call: (9) elder(\_5562, \_5774) ? creep

Exit: (9) elder(charlie, ann) ? creep

Exit: (8) is\_next(charlie) ? creep

X = charlie .

[trace] 5 ?- is\_next(X).

Call: (8) is\_next(\_6508) ? creep

Call: (9) elder(\_6508, \_6720) ? creep

Exit: (9) elder(charlie, ann) ? creep

Exit: (8) is\_next(charlie) ? creep

X = charlie ;

Redo: (9) elder(\_6508, \_6720) ? creep

Exit: (9) elder(ann, andrew) ? creep

Exit: (8) is\_next(ann) ? creep

X = ann ;

Redo: (9) elder(\_6508, \_6720) ? creep

Exit: (9) elder(andrew, edward) ? creep

Exit: (8) is\_next(andrew) ? creep

X = andrew ;

Redo: (8) is\_next(\_6508) ? creep

^ Call: (9) not(elder(\_6508, \_6706)) ? creep

Call: (10) elder(\_6508, \_6706) ? creep

Exit: (10) elder(charlie, ann) ? creep

^ Fail: (9) not(user:elder(\_6508, \_6706)) ? creep

Redo: (8) is\_next(\_6508) ? creep

Call: (9) is\_youngest(\_6508) ? creep

Call: (10) is\_elder(\_6718, \_6508) ? creep

Call: (11) elder(\_6718, \_6508) ? creep

Exit: (11) elder(charlie, ann) ? creep

Exit: (10) is\_elder(charlie, ann) ? creep

Call: (10) is\_elder(\_6718, charlie) ? creep

Call: (11) elder(\_6718, charlie) ? creep

Fail: (11) elder(\_6718, charlie) ? creep

Redo: (10) is\_elder(\_6718, charlie) ? creep

Call: (11) elder(\_6718, \_6720) ? creep

Exit: (11) elder(charlie, ann) ? creep

Call: (11) is\_elder(ann, charlie) ? creep

Call: (12) elder(ann, charlie) ? creep

Fail: (12) elder(ann, charlie) ? creep

Redo: (11) is\_elder(ann, charlie) ? creep

Call: (12) elder(ann, \_6720) ? creep

Exit: (12) elder(ann, andrew) ? creep

Call: (12) is\_elder(andrew, charlie) ? creep

Call: (13) elder(andrew, charlie) ? creep

Fail: (13) elder(andrew, charlie) ? creep

Redo: (12) is\_elder(andrew, charlie) ? creep

Call: (13) elder(andrew, \_6720) ? creep

Exit: (13) elder(andrew, edward) ? creep

Call: (13) is\_elder(edward, charlie) ? creep

Call: (14) elder(edward, charlie) ? creep

Fail: (14) elder(edward, charlie) ? creep

Redo: (13) is\_elder(edward, charlie) ? creep

Call: (14) elder(edward, \_6720) ? creep

Fail: (14) elder(edward, \_6720) ? creep

Fail: (13) is\_elder(edward, charlie) ? creep

Fail: (12) is\_elder(andrew, charlie) ? creep

Fail: (11) is\_elder(ann, charlie) ? creep

Redo: (11) elder(\_6718, \_6720) ? creep

Exit: (11) elder(ann, andrew) ? creep

Call: (11) is\_elder(andrew, charlie) ? creep

Call: (12) elder(andrew, charlie) ? creep

Fail: (12) elder(andrew, charlie) ? creep

Redo: (11) is\_elder(andrew, charlie) ? creep

Call: (12) elder(andrew, \_6720) ? creep

Exit: (12) elder(andrew, edward) ? creep

Call: (12) is\_elder(edward, charlie) ? creep

Call: (13) elder(edward, charlie) ? creep

Fail: (13) elder(edward, charlie) ? creep

Redo: (12) is\_elder(edward, charlie) ? creep

Call: (13) elder(edward, \_6720) ? creep

Fail: (13) elder(edward, \_6720) ? creep

Fail: (12) is\_elder(edward, charlie) ? creep

Fail: (11) is\_elder(andrew, charlie) ? creep

Redo: (11) elder(\_6718, \_6720) ? creep

Exit: (11) elder(andrew, edward) ? creep

Call: (11) is\_elder(edward, charlie) ? creep

Call: (12) elder(edward, charlie) ? creep

Fail: (12) elder(edward, charlie) ? creep

Redo: (11) is\_elder(edward, charlie) ? creep

Call: (12) elder(edward, \_6720) ? creep

Fail: (12) elder(edward, \_6720) ? creep

Fail: (11) is\_elder(edward, charlie) ? creep

Fail: (10) is\_elder(\_6718, charlie) ? creep

Redo: (11) elder(\_6718, \_6508) ? creep

Exit: (11) elder(ann, andrew) ? creep

Exit: (10) is\_elder(ann, andrew) ? creep

Call: (10) is\_elder(\_6718, ann) ? creep

Call: (11) elder(\_6718, ann) ? creep

Exit: (11) elder(charlie, ann) ? creep

Exit: (10) is\_elder(charlie, ann) ? creep

Call: (10) is\_elder(\_6718, charlie) ? creep

Call: (11) elder(\_6718, charlie) ? creep

Fail: (11) elder(\_6718, charlie) ? creep

Redo: (10) is\_elder(\_6718, charlie) ? creep

Call: (11) elder(\_6718, \_6720) ? creep

Exit: (11) elder(charlie, ann) ? creep

Call: (11) is\_elder(ann, charlie) ? creep

Call: (12) elder(ann, charlie) ? creep

Fail: (12) elder(ann, charlie) ? creep

Redo: (11) is\_elder(ann, charlie) ? creep

Call: (12) elder(ann, \_6720) ? creep

Exit: (12) elder(ann, andrew) ? creep

Call: (12) is\_elder(andrew, charlie) ? creep

Call: (13) elder(andrew, charlie) ? creep

Fail: (13) elder(andrew, charlie) ? creep

Redo: (12) is\_elder(andrew, charlie) ? creep

Call: (13) elder(andrew, \_6720) ? creep

Exit: (13) elder(andrew, edward) ? creep

Call: (13) is\_elder(edward, charlie) ? creep

Call: (14) elder(edward, charlie) ? creep

Fail: (14) elder(edward, charlie) ? creep

Redo: (13) is\_elder(edward, charlie) ? creep

Call: (14) elder(edward, \_6720) ? creep

Fail: (14) elder(edward, \_6720) ? creep

Fail: (13) is\_elder(edward, charlie) ? creep

Fail: (12) is\_elder(andrew, charlie) ? creep

Fail: (11) is\_elder(ann, charlie) ? creep

Redo: (11) elder(\_6718, \_6720) ? creep

Exit: (11) elder(ann, andrew) ? creep

Call: (11) is\_elder(andrew, charlie) ? creep

Call: (12) elder(andrew, charlie) ? creep

Fail: (12) elder(andrew, charlie) ? creep

Redo: (11) is\_elder(andrew, charlie) ? creep

Call: (12) elder(andrew, \_6720) ? creep

Exit: (12) elder(andrew, edward) ? creep

Call: (12) is\_elder(edward, charlie) ? creep

Call: (13) elder(edward, charlie) ? creep

Fail: (13) elder(edward, charlie) ? creep

Redo: (12) is\_elder(edward, charlie) ? creep

Call: (13) elder(edward, \_6720) ? creep

Fail: (13) elder(edward, \_6720) ? creep

Fail: (12) is\_elder(edward, charlie) ? creep

Fail: (11) is\_elder(andrew, charlie) ? creep

Redo: (11) elder(\_6718, \_6720) ? creep

Exit: (11) elder(andrew, edward) ? creep

Call: (11) is\_elder(edward, charlie) ? creep

Call: (12) elder(edward, charlie) ? creep

Fail: (12) elder(edward, charlie) ? creep

Redo: (11) is\_elder(edward, charlie) ? creep

Call: (12) elder(edward, \_6720) ? creep

Fail: (12) elder(edward, \_6720) ? creep

Fail: (11) is\_elder(edward, charlie) ? creep

Fail: (10) is\_elder(\_6718, charlie) ? creep

Redo: (10) is\_elder(\_6718, ann) ? creep

Call: (11) elder(\_6718, \_6720) ? creep

Exit: (11) elder(charlie, ann) ? creep

Call: (11) is\_elder(ann, ann) ? creep

Call: (12) elder(ann, ann) ? creep

Fail: (12) elder(ann, ann) ? creep

Redo: (11) is\_elder(ann, ann) ? creep

Call: (12) elder(ann, \_6720) ? creep

Exit: (12) elder(ann, andrew) ? creep

Call: (12) is\_elder(andrew, ann) ? creep

Call: (13) elder(andrew, ann) ? creep

Fail: (13) elder(andrew, ann) ? creep

Redo: (12) is\_elder(andrew, ann) ? creep

Call: (13) elder(andrew, \_6720) ? creep

Exit: (13) elder(andrew, edward) ? creep

Call: (13) is\_elder(edward, ann) ? creep

Call: (14) elder(edward, ann) ? creep

Fail: (14) elder(edward, ann) ? creep

Redo: (13) is\_elder(edward, ann) ? creep

Call: (14) elder(edward, \_6720) ? creep

Fail: (14) elder(edward, \_6720) ? creep

Fail: (13) is\_elder(edward, ann) ? creep

Fail: (12) is\_elder(andrew, ann) ? creep

Fail: (11) is\_elder(ann, ann) ? creep

Redo: (11) elder(\_6718, \_6720) ? creep

Exit: (11) elder(ann, andrew) ? creep

Call: (11) is\_elder(andrew, ann) ? creep

Call: (12) elder(andrew, ann) ? creep

Fail: (12) elder(andrew, ann) ? creep

Redo: (11) is\_elder(andrew, ann) ? creep

Call: (12) elder(andrew, \_6720) ? creep

Exit: (12) elder(andrew, edward) ? creep

Call: (12) is\_elder(edward, ann) ? creep

Call: (13) elder(edward, ann) ? creep

Fail: (13) elder(edward, ann) ? creep

Redo: (12) is\_elder(edward, ann) ? creep

Call: (13) elder(edward, \_6720) ? creep

Fail: (13) elder(edward, \_6720) ? creep

Fail: (12) is\_elder(edward, ann) ? creep

Fail: (11) is\_elder(andrew, ann) ? creep

Redo: (11) elder(\_6718, \_6720) ? creep

Exit: (11) elder(andrew, edward) ? creep

Call: (11) is\_elder(edward, ann) ? creep

Call: (12) elder(edward, ann) ? creep

Fail: (12) elder(edward, ann) ? creep

Redo: (11) is\_elder(edward, ann) ? creep

Call: (12) elder(edward, \_6720) ? creep

Fail: (12) elder(edward, \_6720) ? creep

Fail: (11) is\_elder(edward, ann) ? creep

Fail: (10) is\_elder(\_6718, ann) ? creep

Redo: (11) elder(\_6718, \_6508) ? creep

Exit: (11) elder(andrew, edward) ? creep

Exit: (10) is\_elder(andrew, edward) ? creep

Call: (10) is\_elder(\_6718, andrew) ? creep

Call: (11) elder(\_6718, andrew) ? creep

Exit: (11) elder(ann, andrew) ? creep

Exit: (10) is\_elder(ann, andrew) ? creep

Call: (10) is\_elder(\_6718, ann) ? creep

Call: (11) elder(\_6718, ann) ? creep

Exit: (11) elder(charlie, ann) ? creep

Exit: (10) is\_elder(charlie, ann) ? creep

Exit: (9) is\_youngest(edward) ? creep

Exit: (8) is\_next(edward) ? creep

X = edward .

**Exercise 3: Family Tree**

male(jerry).

male(stuart).

male(warren).

male(peter).

female(kather).

female(maryalice).

female(ann).

brother(jerry,stuart).

brother(jerry,kather).

brother(peter,warren).

sister(ann,maryalice).

sister(kather,jerry).

parent\_of(warren,jerry).

parent\_of(maryalice,jerry).

sibling(X, Y):-

brother(X, Y);

sister(X, Y).

father(X, Y):- male(X), parent\_of(X, Y).

father(X, Y):- male(X), parent\_of(X, Z), sibling(Z, Y).

mother(X, Y):- female(X), parent\_of(X, Y).

mother(X, Y):- female(X), parent\_of(X, Z), sibling(Z, Y).

son(X, Y):- male(X), parent\_of(Y, X).

son(X, Y):- male(X), parent\_of(Y, Z), sibling(Z, X).

daughter(X, Y):- female(X), parent\_of(Y, X).

daughter(X, Y):- female(X), parent\_of(Y, Z), sibling(Z, X).

grandfather(X, Y):- male(X), parent\_of(X, Z), parent\_of(Z, Y).

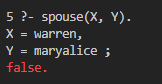
aunt(X, Y):- female(X), sibling(X, Z), parent\_of(Z, Y).

uncle(X, Y):- male(X), sibling(X, Z), parent\_of(Z, Y).

cousin(X, Y):- parent\_of(A, X), parent\_of(B, Y), sibling(A, B).

spouse(X, Y):- parent\_of(X, Z), parent\_of(Y, Z), male(X), female(Y).

parentof(X, Y):- parent\_of(X, Y).



**Trace**

[trace] 2 ?- spouse(X, Y).

Call: (8) spouse(\_3772, \_3774) ? creep

Call: (9) parent\_of(\_3772, \_4032) ? creep

Exit: (9) parent\_of(warren, jerry) ? creep

Call: (9) parent\_of(\_3774, jerry) ? creep

Exit: (9) parent\_of(warren, jerry) ? creep

Call: (9) male(warren) ? creep

Exit: (9) male(warren) ? creep

Call: (9) female(warren) ? creep

Fail: (9) female(warren) ? creep

Redo: (9) parent\_of(\_3774, jerry) ? creep

Exit: (9) parent\_of(maryalice, jerry) ? creep

Call: (9) male(warren) ? creep

Exit: (9) male(warren) ? creep

Call: (9) female(maryalice) ? creep

Exit: (9) female(maryalice) ? creep

Exit: (8) spouse(warren, maryalice) ? creep

X = warren,

Y = maryalice ;

Redo: (9) parent\_of(\_3772, \_4032) ? creep

Exit: (9) parent\_of(maryalice, jerry) ? creep

Call: (9) parent\_of(\_3774, jerry) ? creep

Exit: (9) parent\_of(warren, jerry) ? creep

Call: (9) male(maryalice) ? creep

Fail: (9) male(maryalice) ? creep

Redo: (9) parent\_of(\_3774, jerry) ? creep

Exit: (9) parent\_of(maryalice, jerry) ? creep

Call: (9) male(maryalice) ? creep

Fail: (9) male(maryalice) ? creep

Fail: (8) spouse(\_3772, \_3774) ? creep

false.



[trace] 3 ?-

uncle(X, jerry).

Call: (8) uncle(\_5456, jerry) ? creep

Call: (9) male(\_5456) ? creep

Exit: (9) male(jerry) ? creep

Call: (9) sibling(jerry, \_5676) ? creep

Call: (10) brother(jerry, \_5676) ? creep

Exit: (10) brother(jerry, stuart) ? creep

Exit: (9) sibling(jerry, stuart) ? creep

Call: (9) parent\_of(stuart, jerry) ? creep

Fail: (9) parent\_of(stuart, jerry) ? creep

Redo: (10) brother(jerry, \_5676) ? creep

Exit: (10) brother(jerry, kather) ? creep

Exit: (9) sibling(jerry, kather) ? creep

Call: (9) parent\_of(kather, jerry) ? creep

Fail: (9) parent\_of(kather, jerry) ? creep

Redo: (9) sibling(jerry, \_5676) ? creep

Call: (10) sister(jerry, \_5676) ? creep

Fail: (10) sister(jerry, \_5676) ? creep

Fail: (9) sibling(jerry, \_5676) ? creep

Redo: (9) male(\_5456) ? creep

Exit: (9) male(stuart) ? creep

Call: (9) sibling(stuart, \_5676) ? creep

Call: (10) brother(stuart, \_5676) ? creep

Fail: (10) brother(stuart, \_5676) ? creep

Redo: (9) sibling(stuart, \_5676) ? creep

Call: (10) sister(stuart, \_5676) ? creep

Fail: (10) sister(stuart, \_5676) ? creep

Fail: (9) sibling(stuart, \_5676) ? creep

Redo: (9) male(\_5456) ? creep

Exit: (9) male(warren) ? creep

Call: (9) sibling(warren, \_5676) ? creep

Call: (10) brother(warren, \_5676) ? creep

Fail: (10) brother(warren, \_5676) ? creep

Redo: (9) sibling(warren, \_5676) ? creep

Call: (10) sister(warren, \_5676) ? creep

Fail: (10) sister(warren, \_5676) ? creep

Fail: (9) sibling(warren, \_5676) ? creep

Redo: (9) male(\_5456) ? creep

Exit: (9) male(peter) ? creep

Call: (9) sibling(peter, \_5676) ? creep

Call: (10) brother(peter, \_5676) ? creep

Exit: (10) brother(peter, warren) ? creep

Exit: (9) sibling(peter, warren) ? creep

Call: (9) parent\_of(warren, jerry) ? creep

Exit: (9) parent\_of(warren, jerry) ? creep

Exit: (8) uncle(peter, jerry) ? creep

X = peter ;

Redo: (9) sibling(peter, \_5676) ? creep

Call: (10) sister(peter, \_5676) ? creep

Fail: (10) sister(peter, \_5676) ? creep

Fail: (9) sibling(peter, \_5676) ? creep

Fail: (8) uncle(\_5456, jerry) ? creep

false.