NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

CZ3005 - Artificial Intelligence

Lab 2: Introduction to Prolog

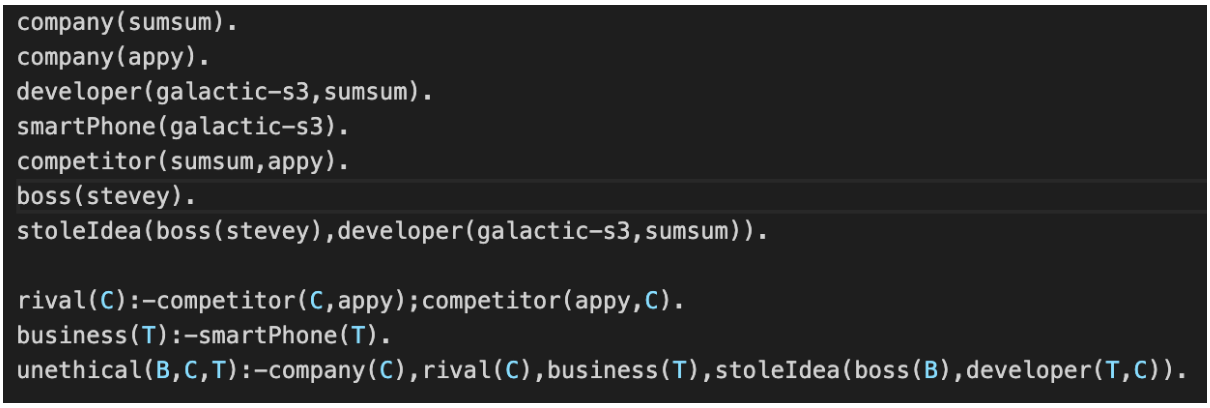
Lab Group: TS7

Vaidyanathan Abhishek. **U1923980D**

QUESTION1 ∀∃∧∨

a) ∀smartPhone(Tech) ⇒ business(Tech)
∀company(X) (∃ X, competitor(X,appy) ∨ competitor(appy,X)) ⇒ rival(X))
∀ company(company1) ∧ rival(company1) ∧ business(Tech) ∧
stoleIdea(boss(person), developer(Tech,company1)) ⇒
unethical(person,company1,Tech)

b)



```
company(sumsum).
company(appy).
developer(galactic-s3,sumsum).
smartPhone(galactic-s3).
competitor(sumsum,appy).
boss(stevey).
stoleIdea(boss(stevey),developer(galactic-s3,sumsum)).

rival(C):-competitor(C,appy);competitor(appy,C).
business(T):-smartPhone(T).
unethical(B,C,T):-company(C),rival(C),business(T),stoleIdea(boss(B),developer(T,C)).
```

c) Trace.

TRACE FOR GENERAL CASE

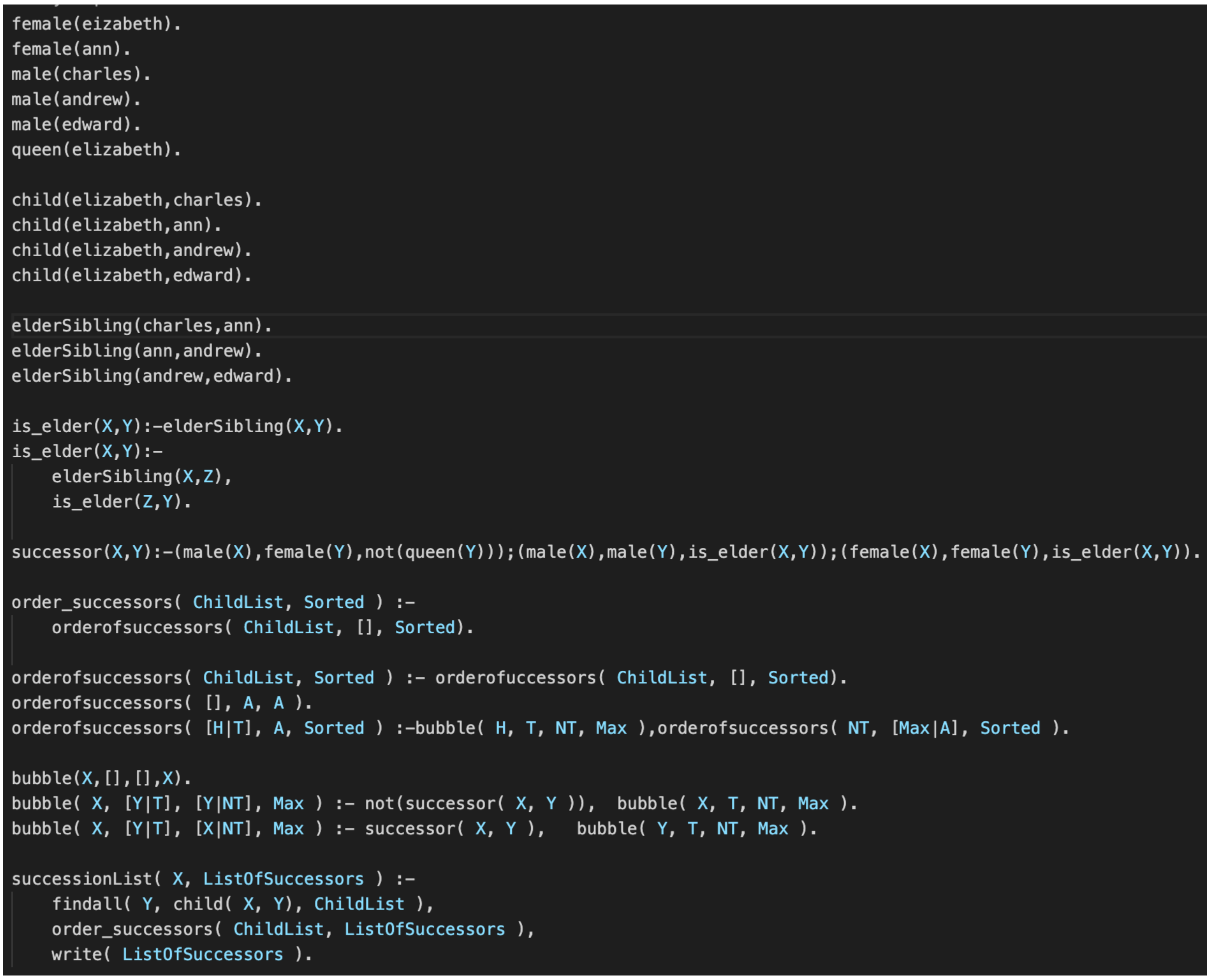
```
?- trace,unethical(A,B,C).
   Call: (11) unethical(_9008, _9010, _9012) ? creep
   Call: (12) company(_9010) ? creep
   Exit: (12) company(sumsum) ? creep
   Call: (12) rival(sumsum) ? creep
   Call: (13) competitor(sumsum, appy) ? creep
   Exit: (13) competitor(sumsum, appy) ? creep
   Exit: (12) rival(sumsum) ? creep
   Call: (12) business(_9012) ? creep
   Call: (13) smartPhone(_9012) ? creep
   Exit: (13) smartPhone(galactic-s3) ? creep
   Exit: (12) business(galactic-s3) ? creep
   Call: (12) stoleIdea(boss(_9008), developer(galactic-s3, sumsum))
? creep
   Exit: (12) stoleIdea(boss(stevey), developer(galactic-s3,
sumsum)) ? creep
   Exit: (11) unethical(stevey, sumsum, galactic-s3) ? creep
A = stevey,
B = sumsum,
C = galactic-s3 .
```

TRACE TO CHECK IF STEVEY IS UNETHICAL

```
?- trace,unethical(stevey,sumsum,galactic-s3).
   Call: (11) unethical(stevey, sumsum, galactic-s3) ? creep
   Call: (12) company(sumsum) ? creep
   Exit: (12) company(sumsum) ? creep
   Call: (12) rival(sumsum) ? creep
   Call: (13) competitor(sumsum, appy) ? creep
   Exit: (13) competitor(sumsum, appy) ? creep
   Exit: (12) rival(sumsum) ? creep
   Call: (12) business(galactic-s3) ? creep
   Call: (13) smartPhone(galactic-s3) ? creep
   Exit: (13) smartPhone(galactic-s3) ? creep
   Exit: (12) business(galactic-s3) ? creep
   Call: (12) stoleIdea(boss(stevey), developer(galactic-s3,
sumsum)) ? creep
   Exit: (12) stoleIdea(boss(stevey), developer(galactic-s3,
sumsum)) ? creep
   Exit: (11) unethical(stevey, sumsum, galactic-s3) ? creep
   true .
```

QUESTION2

a)



```
female(eizabeth).
female(ann).
male(charles).
male(andrew).
male(edward).
queen(elizabeth).

child(elizabeth,charles).
child(elizabeth,ann).
child(elizabeth,andrew).
child(elizabeth,edward).

elderSibling(charles,ann).
elderSibling(ann,andrew).
elderSibling(andrew,edward).

is_elder(X,Y):-elderSibling(X,Y).
is_elder(X,Y):-
    elderSibling(X,Z),
    is_elder(Z,Y).

successor(X,Y):-(male(X),female(Y),not(queen(Y)));(male(X),male(Y),is_elder(X,Y));(female(X),female(Y),is_elder(X,Y)).

order_successors( ChildList, Sorted ) :-
    orderofsuccessors( ChildList, [], Sorted).

orderofsuccessors( ChildList, Sorted ) :- orderofuccessors( ChildList, [], Sorted).
orderofsuccessors( [], A, A ).
orderofsuccessors( [H|T], A, Sorted ) :-bubble( H, T, NT, Max ),orderofsuccessors( NT, [Max|A], Sorted ).

bubble(X,[],[],X).
bubble( X, [Y|T], [Y|NT], Max ) :- not(successor( X, Y )),  bubble( X, T, NT, Max ).
bubble( X, [Y|T], [X|NT], Max ) :- successor( X, Y ),   bubble( Y, T, NT, Max ).

successionList( X, ListOfSuccessors ) :-
    findall( Y, child( X, Y), ChildList ),
    order_successors( ChildList, ListOfSuccessors ),
    write( ListOfSuccessors ).
```

TRACE FOR GENERAL CASE

```
?- trace,successionList(A,B).
   Call: (11) successionList(_9528, _9530) ? creep
^  Call: (12) findall(_10050, child(_9528, _10050), _10110) ? creep
   Call: (17) child(_9528, _10050) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Redo: (17) child(_9528, _10050) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
   Redo: (17) child(_9528, _10050) ? creep
   Exit: (17) child(elizabeth, andrew) ? creep
   Redo: (17) child(_9528, _10050) ? creep
   Exit: (17) child(elizabeth, edward) ? creep
^  Exit: (12) findall(_10050, user:child(_9528, _10050), [charles,
ann, andrew, edward]) ? creep
   Call: (12) order_successors([charles, ann, andrew, edward],
_9530) ? creep
   Call: (13) orderofsuccessors([charles, ann, andrew, edward], [],
_9530) ? creep
   Call: (14) bubble(charles, [ann, andrew, edward], _10696, _10698)
? creep
```

```
^  Call: (15) not(successor(charles, ann)) ? creep
   Call: (16) successor(charles, ann) ? creep
   Call: (17) male(charles) ? creep
   Exit: (17) male(charles) ? creep
   Call: (17) female(ann) ? creep
   Exit: (17) female(ann) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
   Exit: (16) successor(charles, ann) ? creep
^  Fail: (15) not(user:successor(charles, ann)) ? creep
   Redo: (14) bubble(charles, [ann, andrew, edward], _11296, _11298)
? creep
   Call: (15) successor(charles, ann) ? creep
   Call: (16) male(charles) ? creep
   Exit: (16) male(charles) ? creep
   Call: (16) female(ann) ? creep
   Exit: (16) female(ann) ? creep
^  Call: (16) not(queen(ann)) ? creep
   Call: (17) queen(ann) ? creep
   Fail: (17) queen(ann) ? creep
^  Exit: (16) not(user:queen(ann)) ? creep
   Exit: (15) successor(charles, ann) ? creep
   Call: (15) bubble(ann, [andrew, edward], _11286, _11798) ? creep
^  Call: (16) not(successor(ann, andrew)) ? creep
   Call: (17) successor(ann, andrew) ? creep
   Call: (18) male(ann) ? creep
   Fail: (18) male(ann) ? creep
   Redo: (17) successor(ann, andrew) ? creep
   Call: (18) male(ann) ? creep
   Fail: (18) male(ann) ? creep
   Redo: (17) successor(ann, andrew) ? creep
   Call: (18) female(ann) ? creep
   Exit: (18) female(ann) ? creep
   Call: (18) female(andrew) ? creep
   Fail: (18) female(andrew) ? creep
   Fail: (17) successor(ann, andrew) ? creep
^  Exit: (16) not(user:successor(ann, andrew)) ? creep
   Call: (16) bubble(ann, [edward], _11786, _12476) ? creep
^  Call: (17) not(successor(ann, edward)) ? creep
   Call: (18) successor(ann, edward) ? creep
   Call: (19) male(ann) ? creep
   Fail: (19) male(ann) ? creep
   Redo: (18) successor(ann, edward) ? creep
   Call: (19) male(ann) ? creep
   Fail: (19) male(ann) ? creep
   Redo: (18) successor(ann, edward) ? creep
   Call: (19) female(ann) ? creep
   Exit: (19) female(ann) ? creep
   Call: (19) female(edward) ? creep
   Fail: (19) female(edward) ? creep
   Fail: (18) successor(ann, edward) ? creep
^  Exit: (17) not(user:successor(ann, edward)) ? creep
```

```
   Call: (17) bubble(ann, [], _12464, _13154) ? creep
   Exit: (17) bubble(ann, [], [], ann) ? creep
   Exit: (16) bubble(ann, [edward], [edward], ann) ? creep
   Exit: (15) bubble(ann, [andrew, edward], [andrew, edward], ann) ?
creep
   Exit: (14) bubble(charles, [ann, andrew, edward], [charles,
andrew, edward], ann) ? creep
   Call: (14) orderofsuccessors([charles, andrew, edward], [ann],
_9530) ? creep
   Call: (15) bubble(charles, [andrew, edward], _13422, _13424) ?
creep
^  Call: (16) not(successor(charles, andrew)) ? creep
   Call: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) female(andrew) ? creep
   Fail: (18) female(andrew) ? creep
   Redo: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, andrew) ? creep
   Fail: (19) elderSibling(charles, andrew) ? creep
   Redo: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, _14142) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Call: (19) is_elder(ann, andrew) ? creep
   Call: (20) elderSibling(ann, andrew) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Exit: (19) is_elder(ann, andrew) ? creep
   Exit: (18) is_elder(charles, andrew) ? creep
   Exit: (17) successor(charles, andrew) ? creep
^  Fail: (16) not(user:successor(charles, andrew)) ? creep
   Redo: (15) bubble(charles, [andrew, edward], _14540, _14542) ?
creep
   Call: (16) successor(charles, andrew) ? creep
   Call: (17) male(charles) ? creep
   Exit: (17) male(charles) ? creep
   Call: (17) female(andrew) ? creep
   Fail: (17) female(andrew) ? creep
   Redo: (16) successor(charles, andrew) ? creep
   Call: (17) male(charles) ? creep
   Exit: (17) male(charles) ? creep
   Call: (17) male(andrew) ? creep
   Exit: (17) male(andrew) ? creep
   Call: (17) is_elder(charles, andrew) ? creep
   Call: (18) elderSibling(charles, andrew) ? creep
   Fail: (18) elderSibling(charles, andrew) ? creep
   Redo: (17) is_elder(charles, andrew) ? creep
   Call: (18) elderSibling(charles, _15204) ? creep
   Exit: (18) elderSibling(charles, ann) ? creep
   Call: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, andrew) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Exit: (18) is_elder(ann, andrew) ? creep
   Exit: (17) is_elder(charles, andrew) ? creep
   Exit: (16) successor(charles, andrew) ? creep
   Call: (16) bubble(andrew, [edward], _14530, _15560) ? creep
^  Call: (17) not(successor(andrew, edward)) ? creep
   Call: (18) successor(andrew, edward) ? creep
   Call: (19) male(andrew) ? creep
   Exit: (19) male(andrew) ? creep
   Call: (19) female(edward) ? creep
   Fail: (19) female(edward) ? creep
   Redo: (18) successor(andrew, edward) ? creep
   Call: (19) male(andrew) ? creep
   Exit: (19) male(andrew) ? creep
   Call: (19) male(edward) ? creep
   Exit: (19) male(edward) ? creep
   Call: (19) is_elder(andrew, edward) ? creep
   Call: (20) elderSibling(andrew, edward) ? creep
   Exit: (20) elderSibling(andrew, edward) ? creep
   Exit: (19) is_elder(andrew, edward) ? creep
   Exit: (18) successor(andrew, edward) ? creep
^  Fail: (17) not(user:successor(andrew, edward)) ? creep
   Redo: (16) bubble(andrew, [edward], _14530, _16370) ? creep
   Call: (17) successor(andrew, edward) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) female(edward) ? creep
   Fail: (18) female(edward) ? creep
   Redo: (17) successor(andrew, edward) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) male(edward) ? creep
   Exit: (18) male(edward) ? creep
   Call: (18) is_elder(andrew, edward) ? creep
   Call: (19) elderSibling(andrew, edward) ? creep
   Exit: (19) elderSibling(andrew, edward) ? creep
   Exit: (18) is_elder(andrew, edward) ? creep
   Exit: (17) successor(andrew, edward) ? creep
   Call: (17) bubble(edward, [], _16358, _17080) ? creep
   Exit: (17) bubble(edward, [], [], edward) ? creep
   Exit: (16) bubble(andrew, [edward], [andrew], edward) ? creep
   Exit: (15) bubble(charles, [andrew, edward], [charles, andrew],
edward) ? creep
   Call: (15) orderofsuccessors([charles, andrew], [edward, ann],
_9530) ? creep
   Call: (16) bubble(charles, [andrew], _17304, _17306) ? creep
^  Call: (17) not(successor(charles, andrew)) ? creep
   Call: (18) successor(charles, andrew) ? creep
   Call: (19) male(charles) ? creep
   Exit: (19) male(charles) ? creep
   Call: (19) female(andrew) ? creep
   Fail: (19) female(andrew) ? creep
   Redo: (18) successor(charles, andrew) ? creep
   Call: (19) male(charles) ? creep
   Exit: (19) male(charles) ? creep
   Call: (19) male(andrew) ? creep
   Exit: (19) male(andrew) ? creep
   Call: (19) is_elder(charles, andrew) ? creep
   Call: (20) elderSibling(charles, andrew) ? creep
   Fail: (20) elderSibling(charles, andrew) ? creep
   Redo: (19) is_elder(charles, andrew) ? creep
   Call: (20) elderSibling(charles, _18024) ? creep
   Exit: (20) elderSibling(charles, ann) ? creep
   Call: (20) is_elder(ann, andrew) ? creep
   Call: (21) elderSibling(ann, andrew) ? creep
   Exit: (21) elderSibling(ann, andrew) ? creep
   Exit: (20) is_elder(ann, andrew) ? creep
   Exit: (19) is_elder(charles, andrew) ? creep
   Exit: (18) successor(charles, andrew) ? creep
^  Fail: (17) not(user:successor(charles, andrew)) ? creep
   Redo: (16) bubble(charles, [andrew], _18422, _18424) ? creep
   Call: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) female(andrew) ? creep
   Fail: (18) female(andrew) ? creep
   Redo: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, andrew) ? creep
   Fail: (19) elderSibling(charles, andrew) ? creep
   Redo: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, _19086) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Call: (19) is_elder(ann, andrew) ? creep
   Call: (20) elderSibling(ann, andrew) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Exit: (19) is_elder(ann, andrew) ? creep
   Exit: (18) is_elder(charles, andrew) ? creep
   Exit: (17) successor(charles, andrew) ? creep
   Call: (17) bubble(andrew, [], _18412, _19442) ? creep
   Exit: (17) bubble(andrew, [], [], andrew) ? creep
   Exit: (16) bubble(charles, [andrew], [charles], andrew) ? creep
   Call: (16) orderofsuccessors([charles], [andrew, edward, ann],
_9530) ? creep
   Call: (17) bubble(charles, [], _19622, _19624) ? creep
   Exit: (17) bubble(charles, [], [], charles) ? creep
   Call: (17) orderofsuccessors([], [charles, andrew, edward, ann],
_9530) ? creep
   Exit: (17) orderofsuccessors([], [charles, andrew, edward, ann],
[charles, andrew, edward, ann]) ? creep
   Exit: (16) orderofsuccessors([charles], [andrew, edward, ann],
[charles, andrew, edward, ann]) ? creep
```

```
   Exit: (15) orderofsuccessors([charles, andrew], [edward, ann],
[charles, andrew, edward, ann]) ? creep
   Exit: (14) orderofsuccessors([charles, andrew, edward], [ann],
[charles, andrew, edward, ann]) ? creep
   Exit: (13) orderofsuccessors([charles, ann, andrew, edward], [],
[charles, andrew, edward, ann]) ? creep
   Exit: (12) order_successors([charles, ann, andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Call: (12) write([charles, andrew, edward, ann]) ? creep
[charles,andrew,edward,ann]
   Exit: (12) write([charles, andrew, edward, ann]) ? creep
   Exit: (11) successionList(_9528, [charles, andrew, edward, ann])
? creep
B = [charles, andrew, edward, ann] .
```

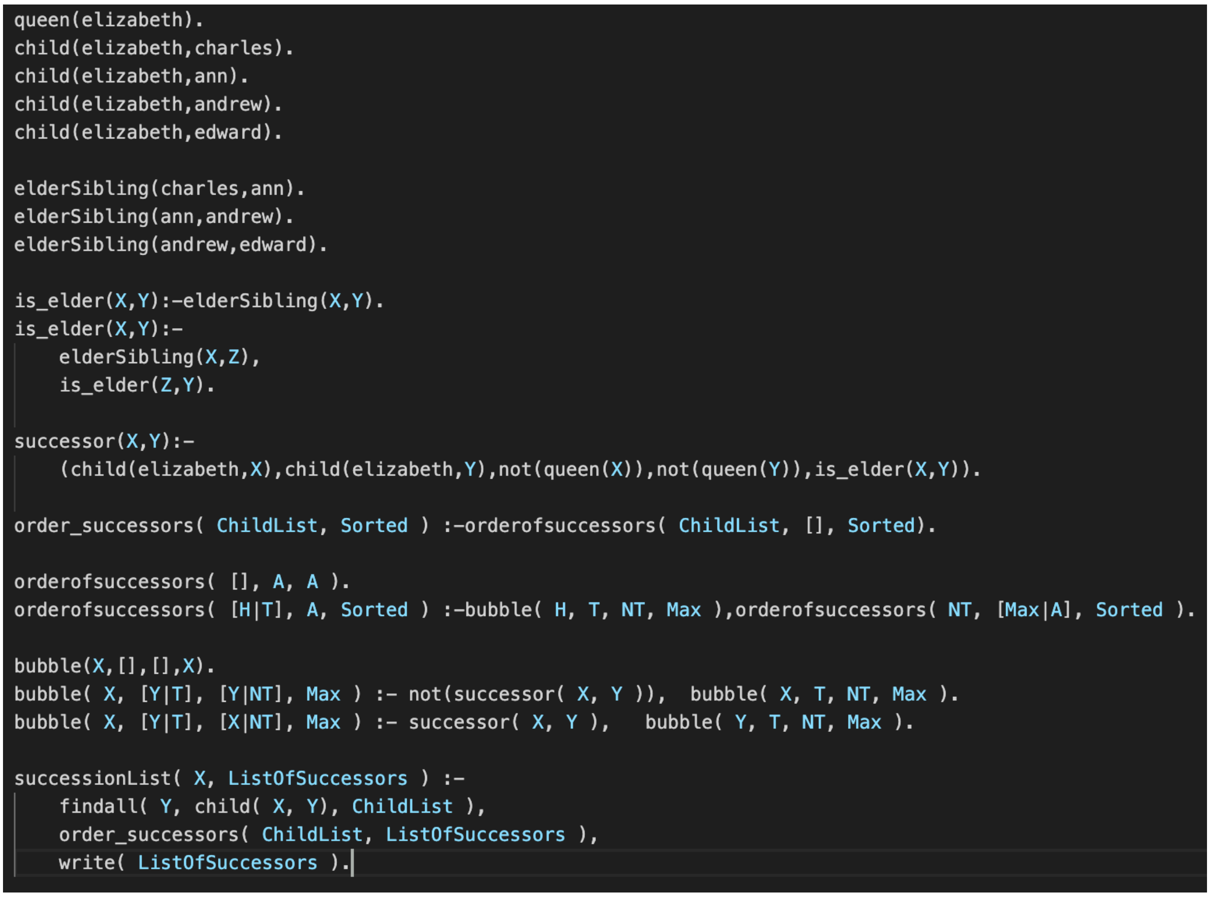
To Check if the successor List Charles, Andrew, Edward, Ann is true

```
?- trace,successionList(elizabeth,[charles,andrew,edward,ann]).
   Call: (11) successionList(elizabeth, [charles, andrew, edward,
ann]) ? creep
^  Call: (12) findall(_9356, child(elizabeth, _9356), _9416) ? creep
   Call: (17) child(elizabeth, _9356) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Redo: (17) child(elizabeth, _9356) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
   Redo: (17) child(elizabeth, _9356) ? creep
   Exit: (17) child(elizabeth, andrew) ? creep
   Redo: (17) child(elizabeth, _9356) ? creep
   Exit: (17) child(elizabeth, edward) ? creep
^  Exit: (12) findall(_9356, user:child(elizabeth, _9356), [charles,
ann, andrew, edward]) ? creep
   Call: (12) order_successors([charles, ann, andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Call: (13) orderofsuccessors([charles, ann, andrew, edward], [],
[charles, andrew, edward, ann]) ? creep
   Call: (14) bubble(charles, [ann, andrew, edward], _10002, _10004)
? creep
^  Call: (15) not(successor(charles, ann)) ? creep
   Call: (16) successor(charles, ann) ? creep
   Call: (17) male(charles) ? creep
   Exit: (17) male(charles) ? creep
   Call: (17) female(ann) ? creep
   Exit: (17) female(ann) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
   Exit: (16) successor(charles, ann) ? creep
^  Fail: (15) not(user:successor(charles, ann)) ? creep
   Redo: (14) bubble(charles, [ann, andrew, edward], _10602, _10604)
? creep
   Call: (15) successor(charles, ann) ? creep
   Call: (16) male(charles) ? creep
   Exit: (16) male(charles) ? creep
   Call: (16) female(ann) ? creep
   Exit: (16) female(ann) ? creep
^  Call: (16) not(queen(ann)) ? creep
   Call: (17) queen(ann) ? creep
   Fail: (17) queen(ann) ? creep
^  Exit: (16) not(user:queen(ann)) ? creep
   Exit: (15) successor(charles, ann) ? creep
   Call: (15) bubble(ann, [andrew, edward], _10592, _11104) ? creep
^  Call: (16) not(successor(ann, andrew)) ? creep
   Call: (17) successor(ann, andrew) ? creep
   Call: (18) male(ann) ? creep
   Fail: (18) male(ann) ? creep
   Redo: (17) successor(ann, andrew) ? creep
   Call: (18) male(ann) ? creep
   Fail: (18) male(ann) ? creep
   Redo: (17) successor(ann, andrew) ? creep
   Call: (18) female(ann) ? creep
   Exit: (18) female(ann) ? creep
   Call: (18) female(andrew) ? creep
   Fail: (18) female(andrew) ? creep
   Fail: (17) successor(ann, andrew) ? creep
^  Exit: (16) not(user:successor(ann, andrew)) ? creep
   Call: (16) bubble(ann, [edward], _11092, _11782) ? creep
^  Call: (17) not(successor(ann, edward)) ? creep
   Call: (18) successor(ann, edward) ? creep
   Call: (19) male(ann) ? creep
   Fail: (19) male(ann) ? creep
   Redo: (18) successor(ann, edward) ? creep
   Call: (19) male(ann) ? creep
   Fail: (19) male(ann) ? creep
   Redo: (18) successor(ann, edward) ? creep
   Call: (19) female(ann) ? creep
   Exit: (19) female(ann) ? creep
   Call: (19) female(edward) ? creep
   Fail: (19) female(edward) ? creep
   Fail: (18) successor(ann, edward) ? creep
^  Exit: (17) not(user:successor(ann, edward)) ? creep
   Call: (17) bubble(ann, [], _11770, _12460) ? creep
   Exit: (17) bubble(ann, [], [], ann) ? creep
   Exit: (16) bubble(ann, [edward], [edward], ann) ? creep
   Exit: (15) bubble(ann, [andrew, edward], [andrew, edward], ann) ?
creep
   Exit: (14) bubble(charles, [ann, andrew, edward], [charles,
andrew, edward], ann) ? creep
   Call: (14) orderofsuccessors([charles, andrew, edward], [ann],
[charles, andrew, edward, ann]) ? creep
   Call: (15) bubble(charles, [andrew, edward], _12728, _12730) ?
creep
^  Call: (16) not(successor(charles, andrew)) ? creep
   Call: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) female(andrew) ? creep
   Fail: (18) female(andrew) ? creep
   Redo: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, andrew) ? creep
   Fail: (19) elderSibling(charles, andrew) ? creep
   Redo: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, _13448) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Call: (19) is_elder(ann, andrew) ? creep
   Call: (20) elderSibling(ann, andrew) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Exit: (19) is_elder(ann, andrew) ? creep
   Exit: (18) is_elder(charles, andrew) ? creep
   Exit: (17) successor(charles, andrew) ? creep
^  Fail: (16) not(user:successor(charles, andrew)) ? creep
   Redo: (15) bubble(charles, [andrew, edward], _13846, _13848) ?
creep
   Call: (16) successor(charles, andrew) ? creep
   Call: (17) male(charles) ? creep
   Exit: (17) male(charles) ? creep
   Call: (17) female(andrew) ? creep
   Fail: (17) female(andrew) ? creep
   Redo: (16) successor(charles, andrew) ? creep
   Call: (17) male(charles) ? creep
   Exit: (17) male(charles) ? creep
   Call: (17) male(andrew) ? creep
   Exit: (17) male(andrew) ? creep
   Call: (17) is_elder(charles, andrew) ? creep
   Call: (18) elderSibling(charles, andrew) ? creep
   Fail: (18) elderSibling(charles, andrew) ? creep
   Redo: (17) is_elder(charles, andrew) ? creep
   Call: (18) elderSibling(charles, _14510) ? creep
   Exit: (18) elderSibling(charles, ann) ? creep
   Call: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, andrew) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Exit: (18) is_elder(ann, andrew) ? creep
   Exit: (17) is_elder(charles, andrew) ? creep
   Exit: (16) successor(charles, andrew) ? creep
   Call: (16) bubble(andrew, [edward], _13836, _14866) ? creep
^  Call: (17) not(successor(andrew, edward)) ? creep
   Call: (18) successor(andrew, edward) ? creep
   Call: (19) male(andrew) ? creep
   Exit: (19) male(andrew) ? creep
   Call: (19) female(edward) ? creep
   Fail: (19) female(edward) ? creep
   Redo: (18) successor(andrew, edward) ? creep
   Call: (19) male(andrew) ? creep
   Exit: (19) male(andrew) ? creep
   Call: (19) male(edward) ? creep
   Exit: (19) male(edward) ? creep
   Call: (19) is_elder(andrew, edward) ? creep
   Call: (20) elderSibling(andrew, edward) ? creep
   Exit: (20) elderSibling(andrew, edward) ? creep
   Exit: (19) is_elder(andrew, edward) ? creep
   Exit: (18) successor(andrew, edward) ? creep
^  Fail: (17) not(user:successor(andrew, edward)) ? creep
   Redo: (16) bubble(andrew, [edward], _13836, _15676) ? creep
   Call: (17) successor(andrew, edward) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) female(edward) ? creep
   Fail: (18) female(edward) ? creep
   Redo: (17) successor(andrew, edward) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) male(edward) ? creep
   Exit: (18) male(edward) ? creep
   Call: (18) is_elder(andrew, edward) ? creep
   Call: (19) elderSibling(andrew, edward) ? creep
   Exit: (19) elderSibling(andrew, edward) ? creep
   Exit: (18) is_elder(andrew, edward) ? creep
   Exit: (17) successor(andrew, edward) ? creep
   Call: (17) bubble(edward, [], _15664, _16386) ? creep
   Exit: (17) bubble(edward, [], [], edward) ? creep
   Exit: (16) bubble(andrew, [edward], [andrew], edward) ? creep
   Exit: (15) bubble(charles, [andrew, edward], [charles, andrew],
edward) ? creep
   Call: (15) orderofsuccessors([charles, andrew], [edward, ann],
[charles, andrew, edward, ann]) ? creep
   Call: (16) bubble(charles, [andrew], _16610, _16612) ? creep
^  Call: (17) not(successor(charles, andrew)) ? creep
   Call: (18) successor(charles, andrew) ? creep
   Call: (19) male(charles) ? creep
   Exit: (19) male(charles) ? creep
   Call: (19) female(andrew) ? creep
   Fail: (19) female(andrew) ? creep
   Redo: (18) successor(charles, andrew) ? creep
   Call: (19) male(charles) ? creep
   Exit: (19) male(charles) ? creep
   Call: (19) male(andrew) ? creep
   Exit: (19) male(andrew) ? creep
   Call: (19) is_elder(charles, andrew) ? creep
   Call: (20) elderSibling(charles, andrew) ? creep
   Fail: (20) elderSibling(charles, andrew) ? creep
   Redo: (19) is_elder(charles, andrew) ? creep
   Call: (20) elderSibling(charles, _17330) ? creep
   Exit: (20) elderSibling(charles, ann) ? creep
   Call: (20) is_elder(ann, andrew) ? creep
   Call: (21) elderSibling(ann, andrew) ? creep
   Exit: (21) elderSibling(ann, andrew) ? creep
   Exit: (20) is_elder(ann, andrew) ? creep
   Exit: (19) is_elder(charles, andrew) ? creep
   Exit: (18) successor(charles, andrew) ? creep
^  Fail: (17) not(user:successor(charles, andrew)) ? creep
   Redo: (16) bubble(charles, [andrew], _17728, _17730) ? creep
   Call: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) female(andrew) ? creep
   Fail: (18) female(andrew) ? creep
   Redo: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, andrew) ? creep
   Fail: (19) elderSibling(charles, andrew) ? creep
   Redo: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, _18392) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Call: (19) is_elder(ann, andrew) ? creep
   Call: (20) elderSibling(ann, andrew) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Exit: (19) is_elder(ann, andrew) ? creep
   Exit: (18) is_elder(charles, andrew) ? creep
   Exit: (17) successor(charles, andrew) ? creep
   Call: (17) bubble(andrew, [], _17718, _18748) ? creep
   Exit: (17) bubble(andrew, [], [], andrew) ? creep
   Exit: (16) bubble(charles, [andrew], [charles], andrew) ? creep
   Call: (16) orderofsuccessors([charles], [andrew, edward, ann],
[charles, andrew, edward, ann]) ? creep
   Call: (17) bubble(charles, [], _18928, _18930) ? creep
   Exit: (17) bubble(charles, [], [], charles) ? creep
   Call: (17) orderofsuccessors([], [charles, andrew, edward, ann],
[charles, andrew, edward, ann]) ? creep
   Exit: (17) orderofsuccessors([], [charles, andrew, edward, ann],
[charles, andrew, edward, ann]) ? creep
   Exit: (16) orderofsuccessors([charles], [andrew, edward, ann],
[charles, andrew, edward, ann]) ? creep
   Exit: (15) orderofsuccessors([charles, andrew], [edward, ann],
[charles, andrew, edward, ann]) ? creep
   Exit: (14) orderofsuccessors([charles, andrew, edward], [ann],
[charles, andrew, edward, ann]) ? creep
   Exit: (13) orderofsuccessors([charles, ann, andrew, edward], [],
[charles, andrew, edward, ann]) ? creep
   Exit: (12) order_successors([charles, ann, andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Call: (12) write([charles, andrew, edward, ann]) ? creep
[charles,andrew,edward,ann]
   Exit: (12) write([charles, andrew, edward, ann]) ? creep
   Exit: (11) successionList(elizabeth, [charles, andrew, edward,
ann]) ? creep
true .
```

To Check if the successor list Charles, Ann, Andrew, Edward is true

```
?- trace,successionList(elizabeth,[charles,ann,andrew,edward]).
   Call: (11) successionList(elizabeth, [charles, ann, andrew, 
edward]) ? creep
^  Call: (12) findall(_4344, child(elizabeth, _4344), _4404) ? creep
   Call: (17) child(elizabeth, _4344) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Redo: (17) child(elizabeth, _4344) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
   Redo: (17) child(elizabeth, _4344) ? creep
   Exit: (17) child(elizabeth, andrew) ? creep
   Redo: (17) child(elizabeth, _4344) ? creep
   Exit: (17) child(elizabeth, edward) ? creep
^  Exit: (12) findall(_4344, user:child(elizabeth, _4344), [charles, 
ann, andrew, edward]) ? creep
   Call: (12) order_successors([charles, ann, andrew, edward], 
[charles, ann, andrew, edward]) ? creep
   Call: (13) orderofsuccessors([charles, ann, andrew, edward], [], 
[charles, ann, andrew, edward]) ? creep
   Call: (14) bubble(charles, [ann, andrew, edward], _4990, _4992) ? 
creep
^  Call: (15) not(successor(charles, ann)) ? creep
   Call: (16) successor(charles, ann) ? creep
   Call: (17) male(charles) ? creep
   Exit: (17) male(charles) ? creep
   Call: (17) female(ann) ? creep
   Exit: (17) female(ann) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
   Exit: (16) successor(charles, ann) ? creep
^  Fail: (15) not(user:successor(charles, ann)) ? creep
   Redo: (14) bubble(charles, [ann, andrew, edward], _5590, _5592) ? 
creep
   Call: (15) successor(charles, ann) ? creep
   Call: (16) male(charles) ? creep
   Exit: (16) male(charles) ? creep
   Call: (16) female(ann) ? creep
   Exit: (16) female(ann) ? creep
^  Call: (16) not(queen(ann)) ? creep
   Call: (17) queen(ann) ? creep
   Fail: (17) queen(ann) ? creep
^  Exit: (16) not(user:queen(ann)) ? creep
   Exit: (15) successor(charles, ann) ? creep
   Call: (15) bubble(ann, [andrew, edward], _5580, _6092) ? creep
^  Call: (16) not(successor(ann, andrew)) ? creep
   Call: (17) successor(ann, andrew) ? creep
   Call: (18) male(ann) ? creep
   Fail: (18) male(ann) ? creep
   Redo: (17) successor(ann, andrew) ? creep
   Call: (18) male(ann) ? creep
   Fail: (18) male(ann) ? creep
   Redo: (17) successor(ann, andrew) ? creep
   Call: (18) female(ann) ? creep
   Exit: (18) female(ann) ? creep
   Call: (18) female(andrew) ? creep
   Fail: (18) female(andrew) ? creep
   Fail: (17) successor(ann, andrew) ? creep
^  Exit: (16) not(user:successor(ann, andrew)) ? creep
   Call: (16) bubble(ann, [edward], _6080, _6770) ? creep
^  Call: (17) not(successor(ann, edward)) ? creep
   Call: (18) successor(ann, edward) ? creep
   Call: (19) male(ann) ? creep
   Fail: (19) male(ann) ? creep
   Redo: (18) successor(ann, edward) ? creep
   Call: (19) male(ann) ? creep
   Fail: (19) male(ann) ? creep
   Redo: (18) successor(ann, edward) ? creep
   Call: (19) female(ann) ? creep
   Exit: (19) female(ann) ? creep
   Call: (19) female(edward) ? creep
   Fail: (19) female(edward) ? creep
   Fail: (18) successor(ann, edward) ? creep
^  Exit: (17) not(user:successor(ann, edward)) ? creep
   Call: (17) bubble(ann, [], _6758, _7448) ? creep
   Exit: (17) bubble(ann, [], [], ann) ? creep
   Exit: (16) bubble(ann, [edward], [edward], ann) ? creep
   Exit: (15) bubble(ann, [andrew, edward], [andrew, edward], ann) ?
creep
   Exit: (14) bubble(charles, [ann, andrew, edward], [charles,
andrew, edward], ann) ? creep
   Call: (14) orderofsuccessors([charles, andrew, edward], [ann],
[charles, ann, andrew, edward]) ? creep
   Call: (15) bubble(charles, [andrew, edward], _7716, _7718) ?
creep
^  Call: (16) not(successor(charles, andrew)) ? creep
   Call: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) female(andrew) ? creep
   Fail: (18) female(andrew) ? creep
   Redo: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, andrew) ? creep
   Fail: (19) elderSibling(charles, andrew) ? creep
   Redo: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, _8436) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Call: (19) is_elder(ann, andrew) ? creep
   Call: (20) elderSibling(ann, andrew) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Exit: (19) is_elder(ann, andrew) ? creep
   Exit: (18) is_elder(charles, andrew) ? creep
   Exit: (17) successor(charles, andrew) ? creep
^  Fail: (16) not(user:successor(charles, andrew)) ? creep
   Redo: (15) bubble(charles, [andrew, edward], _8834, _8836) ?
creep
   Call: (16) successor(charles, andrew) ? creep
   Call: (17) male(charles) ? creep
   Exit: (17) male(charles) ? creep
   Call: (17) female(andrew) ? creep
   Fail: (17) female(andrew) ? creep
   Redo: (16) successor(charles, andrew) ? creep
   Call: (17) male(charles) ? creep
   Exit: (17) male(charles) ? creep
   Call: (17) male(andrew) ? creep
   Exit: (17) male(andrew) ? creep
   Call: (17) is_elder(charles, andrew) ? creep
   Call: (18) elderSibling(charles, andrew) ? creep
   Fail: (18) elderSibling(charles, andrew) ? creep
   Redo: (17) is_elder(charles, andrew) ? creep
   Call: (18) elderSibling(charles, _9498) ? creep
   Exit: (18) elderSibling(charles, ann) ? creep
   Call: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, andrew) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Exit: (18) is_elder(ann, andrew) ? creep
   Exit: (17) is_elder(charles, andrew) ? creep
   Exit: (16) successor(charles, andrew) ? creep
   Call: (16) bubble(andrew, [edward], _8824, _9854) ? creep
^  Call: (17) not(successor(andrew, edward)) ? creep
   Call: (18) successor(andrew, edward) ? creep
   Call: (19) male(andrew) ? creep
   Exit: (19) male(andrew) ? creep
   Call: (19) female(edward) ? creep
   Fail: (19) female(edward) ? creep
   Redo: (18) successor(andrew, edward) ? creep
   Call: (19) male(andrew) ? creep
   Exit: (19) male(andrew) ? creep
   Call: (19) male(edward) ? creep
   Exit: (19) male(edward) ? creep
   Call: (19) is_elder(andrew, edward) ? creep
   Call: (20) elderSibling(andrew, edward) ? creep
   Exit: (20) elderSibling(andrew, edward) ? creep
   Exit: (19) is_elder(andrew, edward) ? creep
   Exit: (18) successor(andrew, edward) ? creep
^  Fail: (17) not(user:successor(andrew, edward)) ? creep
   Redo: (16) bubble(andrew, [edward], _8824, _10664) ? creep
   Call: (17) successor(andrew, edward) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) female(edward) ? creep
   Fail: (18) female(edward) ? creep
   Redo: (17) successor(andrew, edward) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) male(edward) ? creep
   Exit: (18) male(edward) ? creep
   Call: (18) is_elder(andrew, edward) ? creep
   Call: (19) elderSibling(andrew, edward) ? creep
   Exit: (19) elderSibling(andrew, edward) ? creep
   Exit: (18) is_elder(andrew, edward) ? creep
   Exit: (17) successor(andrew, edward) ? creep
   Call: (17) bubble(edward, [], _10652, _11374) ? creep
   Exit: (17) bubble(edward, [], [], edward) ? creep
   Exit: (16) bubble(andrew, [edward], [andrew], edward) ? creep
   Exit: (15) bubble(charles, [andrew, edward], [charles, andrew],
edward) ? creep
   Call: (15) orderofsuccessors([charles, andrew], [edward, ann],
[charles, ann, andrew, edward]) ? creep
   Call: (16) bubble(charles, [andrew], _11598, _11600) ? creep
^  Call: (17) not(successor(charles, andrew)) ? creep
   Call: (18) successor(charles, andrew) ? creep
   Call: (19) male(charles) ? creep
   Exit: (19) male(charles) ? creep
   Call: (19) female(andrew) ? creep
   Fail: (19) female(andrew) ? creep
   Redo: (18) successor(charles, andrew) ? creep
   Call: (19) male(charles) ? creep
   Exit: (19) male(charles) ? creep
   Call: (19) male(andrew) ? creep
   Exit: (19) male(andrew) ? creep
   Call: (19) is_elder(charles, andrew) ? creep
   Call: (20) elderSibling(charles, andrew) ? creep
   Fail: (20) elderSibling(charles, andrew) ? creep
   Redo: (19) is_elder(charles, andrew) ? creep
   Call: (20) elderSibling(charles, _12318) ? creep
   Exit: (20) elderSibling(charles, ann) ? creep
   Call: (20) is_elder(ann, andrew) ? creep
   Call: (21) elderSibling(ann, andrew) ? creep
   Exit: (21) elderSibling(ann, andrew) ? creep
   Exit: (20) is_elder(ann, andrew) ? creep
   Exit: (19) is_elder(charles, andrew) ? creep
   Exit: (18) successor(charles, andrew) ? creep
^  Fail: (17) not(user:successor(charles, andrew)) ? creep
   Redo: (16) bubble(charles, [andrew], _12716, _12718) ? creep
   Call: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) female(andrew) ? creep
   Fail: (18) female(andrew) ? creep
   Redo: (17) successor(charles, andrew) ? creep
   Call: (18) male(charles) ? creep
   Exit: (18) male(charles) ? creep
   Call: (18) male(andrew) ? creep
   Exit: (18) male(andrew) ? creep
   Call: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, andrew) ? creep
   Fail: (19) elderSibling(charles, andrew) ? creep
   Redo: (18) is_elder(charles, andrew) ? creep
   Call: (19) elderSibling(charles, _13380) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Call: (19) is_elder(ann, andrew) ? creep
   Call: (20) elderSibling(ann, andrew) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Exit: (19) is_elder(ann, andrew) ? creep
   Exit: (18) is_elder(charles, andrew) ? creep
   Exit: (17) successor(charles, andrew) ? creep
   Call: (17) bubble(andrew, [], _12706, _13736) ? creep
   Exit: (17) bubble(andrew, [], [], andrew) ? creep
   Exit: (16) bubble(charles, [andrew], [charles], andrew) ? creep
   Call: (16) orderofsuccessors([charles], [andrew, edward, ann],
[charles, ann, andrew, edward]) ? creep
   Call: (17) bubble(charles, [], _13916, _13918) ? creep
   Exit: (17) bubble(charles, [], [], charles) ? creep
   Call: (17) orderofsuccessors([], [charles, andrew, edward, ann],
[charles, ann, andrew, edward]) ? creep
   Fail: (17) orderofsuccessors([], [charles, andrew, edward, ann],
[charles, ann, andrew, edward]) ? creep
   Redo: (17) bubble(charles, [], _14098, _14100) ? creep
   Fail: (17) bubble(charles, [], _14142, _14144) ? creep
   Fail: (16) orderofsuccessors([charles], [andrew, edward, ann],
[charles, ann, andrew, edward]) ? creep
   Redo: (17) bubble(andrew, [], _12706, _14232) ? creep
   Fail: (17) bubble(andrew, [], _12706, _14276) ? creep
   Redo: (19) is_elder(ann, andrew) ? creep
   Call: (20) elderSibling(ann, _14360) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Call: (20) is_elder(andrew, andrew) ? creep
   Call: (21) elderSibling(andrew, andrew) ? creep
   Fail: (21) elderSibling(andrew, andrew) ? creep
   Redo: (20) is_elder(andrew, andrew) ? creep
   Call: (21) elderSibling(andrew, _14624) ? creep
   Exit: (21) elderSibling(andrew, edward) ? creep
   Call: (21) is_elder(edward, andrew) ? creep
   Call: (22) elderSibling(edward, andrew) ? creep
   Fail: (22) elderSibling(edward, andrew) ? creep
   Redo: (21) is_elder(edward, andrew) ? creep
   Call: (22) elderSibling(edward, _14888) ? creep
   Fail: (22) elderSibling(edward, _14932) ? creep
   Fail: (21) is_elder(edward, andrew) ? creep
   Fail: (20) is_elder(andrew, andrew) ? creep
   Fail: (19) is_elder(ann, andrew) ? creep
   Fail: (18) is_elder(charles, andrew) ? creep
   Redo: (17) successor(charles, andrew) ? creep
   Call: (18) female(charles) ? creep
   Fail: (18) female(charles) ? creep
   Fail: (17) successor(charles, andrew) ? creep
   Fail: (16) bubble(charles, [andrew], _15330, _15332) ? creep
   Fail: (15) orderofsuccessors([charles, andrew], [edward, ann],
[charles, ann, andrew, edward]) ? creep
   Redo: (17) bubble(edward, [], _10652, _15420) ? creep
   Fail: (17) bubble(edward, [], _10652, _15464) ? creep
   Redo: (18) is_elder(andrew, edward) ? creep
   Call: (19) elderSibling(andrew, _15548) ? creep
   Exit: (19) elderSibling(andrew, edward) ? creep
   Call: (19) is_elder(edward, edward) ? creep
   Call: (20) elderSibling(edward, edward) ? creep
   Fail: (20) elderSibling(edward, edward) ? creep
   Redo: (19) is_elder(edward, edward) ? creep
   Call: (20) elderSibling(edward, _15812) ? creep
   Fail: (20) elderSibling(edward, _15856) ? creep
   Fail: (19) is_elder(edward, edward) ? creep
   Fail: (18) is_elder(andrew, edward) ? creep
   Redo: (17) successor(andrew, edward) ? creep
   Call: (18) female(andrew) ? creep
   Fail: (18) female(andrew) ? creep
   Fail: (17) successor(andrew, edward) ? creep
   Fail: (16) bubble(andrew, [edward], _8824, _16168) ? creep
   Redo: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, _16252) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Call: (19) is_elder(andrew, andrew) ? creep
   Call: (20) elderSibling(andrew, andrew) ? creep
   Fail: (20) elderSibling(andrew, andrew) ? creep
   Redo: (19) is_elder(andrew, andrew) ? creep
   Call: (20) elderSibling(andrew, _16516) ? creep
   Exit: (20) elderSibling(andrew, edward) ? creep
   Call: (20) is_elder(edward, andrew) ? creep
   Call: (21) elderSibling(edward, andrew) ? creep
   Fail: (21) elderSibling(edward, andrew) ? creep
   Redo: (20) is_elder(edward, andrew) ? creep
   Call: (21) elderSibling(edward, _16780) ? creep
   Fail: (21) elderSibling(edward, _16824) ? creep
   Fail: (20) is_elder(edward, andrew) ? creep
   Fail: (19) is_elder(andrew, andrew) ? creep
   Fail: (18) is_elder(ann, andrew) ? creep
   Fail: (17) is_elder(charles, andrew) ? creep
   Redo: (16) successor(charles, andrew) ? creep
   Call: (17) female(charles) ? creep
   Fail: (17) female(charles) ? creep
   Fail: (16) successor(charles, andrew) ? creep
   Fail: (15) bubble(charles, [andrew, edward], _17222, _17224) ?
creep
   Fail: (14) orderofsuccessors([charles, andrew, edward], [ann],
[charles, ann, andrew, edward]) ? creep
   Redo: (17) bubble(ann, [], _6758, _17312) ? creep
   Fail: (17) bubble(ann, [], _6758, _17356) ? creep
   Redo: (16) bubble(ann, [edward], _6080, _17400) ? creep
   Call: (17) successor(ann, edward) ? creep
   Call: (18) male(ann) ? creep
   Fail: (18) male(ann) ? creep
   Redo: (17) successor(ann, edward) ? creep
   Call: (18) male(ann) ? creep
   Fail: (18) male(ann) ? creep
   Redo: (17) successor(ann, edward) ? creep
   Call: (18) female(ann) ? creep
   Exit: (18) female(ann) ? creep
   Call: (18) female(edward) ? creep
   Fail: (18) female(edward) ? creep
   Fail: (17) successor(ann, edward) ? creep
   Fail: (16) bubble(ann, [edward], _6080, _17978) ? creep
   Redo: (15) bubble(ann, [andrew, edward], _5580, _18022) ? creep
   Call: (16) successor(ann, andrew) ? creep
   Call: (17) male(ann) ? creep
   Fail: (17) male(ann) ? creep
   Redo: (16) successor(ann, andrew) ? creep
   Call: (17) male(ann) ? creep
   Fail: (17) male(ann) ? creep
   Redo: (16) successor(ann, andrew) ? creep
   Call: (17) female(ann) ? creep
   Exit: (17) female(ann) ? creep
   Call: (17) female(andrew) ? creep
   Fail: (17) female(andrew) ? creep
   Fail: (16) successor(ann, andrew) ? creep
   Fail: (15) bubble(ann, [andrew, edward], _5580, _18600) ? creep
   Redo: (15) successor(charles, ann) ? creep
   Call: (16) male(charles) ? creep
   Exit: (16) male(charles) ? creep
   Call: (16) male(ann) ? creep
   Fail: (16) male(ann) ? creep
   Redo: (15) successor(charles, ann) ? creep
   Call: (16) female(charles) ? creep
   Fail: (16) female(charles) ? creep
   Fail: (15) successor(charles, ann) ? creep
   Fail: (14) bubble(charles, [ann, andrew, edward], _19038, _19040)
? creep
   Fail: (13) orderofsuccessors([charles, ann, andrew, edward], [],
[charles, ann, andrew, edward]) ? creep
   Fail: (12) order_successors([charles, ann, andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Fail: (11) successionList(elizabeth, [charles, ann, andrew,
edward]) ? creep
false.
```

b) the necessary changes are to the knowledge base which don’t require the information of whether the successor is male or female. The only fact that we should check is that they shouldn’t be the queen. Also point to be noted is that the rule for to be the new successor is only dependent on whether or not he/she is older than the other person. Bubble sorting changes accordingly.



```
queen(elizabeth).
child(elizabeth,charles).
child(elizabeth,ann).
child(elizabeth,andrew).
child(elizabeth,edward).

elderSibling(charles,ann).
elderSibling(ann,andrew).
elderSibling(andrew,edward).

is_elder(X,Y):-elderSibling(X,Y).
is_elder(X,Y):-
    elderSibling(X,Z),
    is_elder(Z,Y).

successor(X,Y):-
    (child(elizabeth,X),child(elizabeth,Y),not(queen(X)),not(queen(Y)),is_elder(X,Y)).

order_successors( ChildList, Sorted ) :-orderofsuccessors( ChildList, [], Sorted).

orderofsuccessors( [], A, A ).
orderofsuccessors( [H|T], A, Sorted ) :-bubble( H, T, NT, Max ),orderofsuccessors( NT, [Max|A], Sorted ).

bubble(X,[],[],X).
bubble( X, [Y|T], [Y|NT], Max ) :- not(successor( X, Y )),  bubble( X, T, NT, Max ).
bubble( X, [Y|T], [X|NT], Max ) :- successor( X, Y ),   bubble( Y, T, NT, Max ).

successionList( X, ListOfSuccessors ) :-
    findall( Y, child( X, Y), ChildList ),
    order_successors( ChildList, ListOfSuccessors ),
    write( ListOfSuccessors ).
```

TRACE FOR GENERAL CASE

```
?- trace, successionList(A,B).
   Call: (11) successionList(_3818, _3820) ? creep
^  Call: (12) findall(_4340, child(_3818, _4340), _4400) ? creep
   Call: (17) child(_3818, _4340) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Redo: (17) child(_3818, _4340) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
   Redo: (17) child(_3818, _4340) ? creep
   Exit: (17) child(elizabeth, andrew) ? creep
   Redo: (17) child(_3818, _4340) ? creep
   Exit: (17) child(elizabeth, edward) ? creep
^  Exit: (12) findall(_4340, user:child(_3818, _4340), [charles,
ann, andrew, edward]) ? creep
   Call: (12) order_successors([charles, ann, andrew, edward],
_3820) ? creep
   Call: (13) orderofsuccessors([charles, ann, andrew, edward], [],
_3820) ? creep
   Call: (14) bubble(charles, [ann, andrew, edward], _4986, _4988) ?
creep
^  Call: (15) not(successor(charles, ann)) ? creep
   Call: (16) successor(charles, ann) ? creep
   Call: (17) child(elizabeth, charles) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Call: (17) child(elizabeth, ann) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
^  Call: (17) not(queen(charles)) ? creep
   Call: (18) queen(charles) ? creep
   Fail: (18) queen(charles) ? creep
^  Exit: (17) not(user:queen(charles)) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
   Call: (17) is_elder(charles, ann) ? creep
   Call: (18) elderSibling(charles, ann) ? creep
   Exit: (18) elderSibling(charles, ann) ? creep
   Exit: (17) is_elder(charles, ann) ? creep
   Exit: (16) successor(charles, ann) ? creep
^  Fail: (15) not(user:successor(charles, ann)) ? creep
   Redo: (14) bubble(charles, [ann, andrew, edward], _5948, _5950) ?
creep
   Call: (15) successor(charles, ann) ? creep
   Call: (16) child(elizabeth, charles) ? creep
   Exit: (16) child(elizabeth, charles) ? creep
   Call: (16) child(elizabeth, ann) ? creep
   Exit: (16) child(elizabeth, ann) ? creep
^  Call: (16) not(queen(charles)) ? creep
   Call: (17) queen(charles) ? creep
   Fail: (17) queen(charles) ? creep
^  Exit: (16) not(user:queen(charles)) ? creep
^  Call: (16) not(queen(ann)) ? creep
   Call: (17) queen(ann) ? creep
   Fail: (17) queen(ann) ? creep
^  Exit: (16) not(user:queen(ann)) ? creep
   Call: (16) is_elder(charles, ann) ? creep
   Call: (17) elderSibling(charles, ann) ? creep
   Exit: (17) elderSibling(charles, ann) ? creep
   Exit: (16) is_elder(charles, ann) ? creep
   Exit: (15) successor(charles, ann) ? creep
   Call: (15) bubble(ann, [andrew, edward], _5938, _6812) ? creep
^  Call: (16) not(successor(ann, andrew)) ? creep
   Call: (17) successor(ann, andrew) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
   Call: (18) child(elizabeth, andrew) ? creep
   Exit: (18) child(elizabeth, andrew) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
^  Call: (18) not(queen(andrew)) ? creep
   Call: (19) queen(andrew) ? creep
   Fail: (19) queen(andrew) ? creep
^  Exit: (18) not(user:queen(andrew)) ? creep
   Call: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, andrew) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Exit: (18) is_elder(ann, andrew) ? creep
   Exit: (17) successor(ann, andrew) ? creep
^  Fail: (16) not(user:successor(ann, andrew)) ? creep
   Redo: (15) bubble(ann, [andrew, edward], _5938, _7774) ? creep
   Call: (16) successor(ann, andrew) ? creep
   Call: (17) child(elizabeth, ann) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
   Call: (17) child(elizabeth, andrew) ? creep
   Exit: (17) child(elizabeth, andrew) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
^  Call: (17) not(queen(andrew)) ? creep
   Call: (18) queen(andrew) ? creep
   Fail: (18) queen(andrew) ? creep
^  Exit: (17) not(user:queen(andrew)) ? creep
   Call: (17) is_elder(ann, andrew) ? creep
   Call: (18) elderSibling(ann, andrew) ? creep
   Exit: (18) elderSibling(ann, andrew) ? creep
   Exit: (17) is_elder(ann, andrew) ? creep
   Exit: (16) successor(ann, andrew) ? creep
   Call: (16) bubble(andrew, [edward], _7762, _8636) ? creep
^  Call: (17) not(successor(andrew, edward)) ? creep
   Call: (18) successor(andrew, edward) ? creep
   Call: (19) child(elizabeth, andrew) ? creep
   Exit: (19) child(elizabeth, andrew) ? creep
   Call: (19) child(elizabeth, edward) ? creep
   Exit: (19) child(elizabeth, edward) ? creep
^  Call: (19) not(queen(andrew)) ? creep
   Call: (20) queen(andrew) ? creep
   Fail: (20) queen(andrew) ? creep
^  Exit: (19) not(user:queen(andrew)) ? creep
^  Call: (19) not(queen(edward)) ? creep
   Call: (20) queen(edward) ? creep
   Fail: (20) queen(edward) ? creep
^  Exit: (19) not(user:queen(edward)) ? creep
   Call: (19) is_elder(andrew, edward) ? creep
   Call: (20) elderSibling(andrew, edward) ? creep
   Exit: (20) elderSibling(andrew, edward) ? creep
   Exit: (19) is_elder(andrew, edward) ? creep
   Exit: (18) successor(andrew, edward) ? creep
^  Fail: (17) not(user:successor(andrew, edward)) ? creep
   Redo: (16) bubble(andrew, [edward], _7762, _9598) ? creep
   Call: (17) successor(andrew, edward) ? creep
   Call: (18) child(elizabeth, andrew) ? creep
   Exit: (18) child(elizabeth, andrew) ? creep
   Call: (18) child(elizabeth, edward) ? creep
   Exit: (18) child(elizabeth, edward) ? creep
^  Call: (18) not(queen(andrew)) ? creep
   Call: (19) queen(andrew) ? creep
   Fail: (19) queen(andrew) ? creep
^  Exit: (18) not(user:queen(andrew)) ? creep
^  Call: (18) not(queen(edward)) ? creep
   Call: (19) queen(edward) ? creep
   Fail: (19) queen(edward) ? creep
^  Exit: (18) not(user:queen(edward)) ? creep
   Call: (18) is_elder(andrew, edward) ? creep
   Call: (19) elderSibling(andrew, edward) ? creep
   Exit: (19) elderSibling(andrew, edward) ? creep
   Exit: (18) is_elder(andrew, edward) ? creep
   Exit: (17) successor(andrew, edward) ? creep
   Call: (17) bubble(edward, [], _9586, _10460) ? creep
   Exit: (17) bubble(edward, [], [], edward) ? creep
   Exit: (16) bubble(andrew, [edward], [andrew], edward) ? creep
   Exit: (15) bubble(ann, [andrew, edward], [ann, andrew], edward) ?
creep
   Exit: (14) bubble(charles, [ann, andrew, edward], [charles, ann,
andrew], edward) ? creep
   Call: (14) orderofsuccessors([charles, ann, andrew], [edward],
_3820) ? creep
   Call: (15) bubble(charles, [ann, andrew], _10728, _10730) ? creep
^  Call: (16) not(successor(charles, ann)) ? creep
   Call: (17) successor(charles, ann) ? creep
   Call: (18) child(elizabeth, charles) ? creep
   Exit: (18) child(elizabeth, charles) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
^  Call: (18) not(queen(charles)) ? creep
   Call: (19) queen(charles) ? creep
   Fail: (19) queen(charles) ? creep
^  Exit: (18) not(user:queen(charles)) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
   Call: (18) is_elder(charles, ann) ? creep
   Call: (19) elderSibling(charles, ann) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Exit: (18) is_elder(charles, ann) ? creep
   Exit: (17) successor(charles, ann) ? creep
^  Fail: (16) not(user:successor(charles, ann)) ? creep
   Redo: (15) bubble(charles, [ann, andrew], _11690, _11692) ? creep
   Call: (16) successor(charles, ann) ? creep
   Call: (17) child(elizabeth, charles) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Call: (17) child(elizabeth, ann) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
^  Call: (17) not(queen(charles)) ? creep
   Call: (18) queen(charles) ? creep
   Fail: (18) queen(charles) ? creep
^  Exit: (17) not(user:queen(charles)) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
   Call: (17) is_elder(charles, ann) ? creep
   Call: (18) elderSibling(charles, ann) ? creep
   Exit: (18) elderSibling(charles, ann) ? creep
   Exit: (17) is_elder(charles, ann) ? creep
   Exit: (16) successor(charles, ann) ? creep
   Call: (16) bubble(ann, [andrew], _11680, _12554) ? creep
^  Call: (17) not(successor(ann, andrew)) ? creep
   Call: (18) successor(ann, andrew) ? creep
   Call: (19) child(elizabeth, ann) ? creep
   Exit: (19) child(elizabeth, ann) ? creep
   Call: (19) child(elizabeth, andrew) ? creep
   Exit: (19) child(elizabeth, andrew) ? creep
^  Call: (19) not(queen(ann)) ? creep
   Call: (20) queen(ann) ? creep
   Fail: (20) queen(ann) ? creep
^  Exit: (19) not(user:queen(ann)) ? creep
^  Call: (19) not(queen(andrew)) ? creep
   Call: (20) queen(andrew) ? creep
   Fail: (20) queen(andrew) ? creep
^  Exit: (19) not(user:queen(andrew)) ? creep
   Call: (19) is_elder(ann, andrew) ? creep
   Call: (20) elderSibling(ann, andrew) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Exit: (19) is_elder(ann, andrew) ? creep
   Exit: (18) successor(ann, andrew) ? creep
^  Fail: (17) not(user:successor(ann, andrew)) ? creep
   Redo: (16) bubble(ann, [andrew], _11680, _13516) ? creep
   Call: (17) successor(ann, andrew) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
   Call: (18) child(elizabeth, andrew) ? creep
   Exit: (18) child(elizabeth, andrew) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
^  Call: (18) not(queen(andrew)) ? creep
   Call: (19) queen(andrew) ? creep
   Fail: (19) queen(andrew) ? creep
^  Exit: (18) not(user:queen(andrew)) ? creep
   Call: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, andrew) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Exit: (18) is_elder(ann, andrew) ? creep
   Exit: (17) successor(ann, andrew) ? creep
   Call: (17) bubble(andrew, [], _13504, _14378) ? creep
   Exit: (17) bubble(andrew, [], [], andrew) ? creep
   Exit: (16) bubble(ann, [andrew], [ann], andrew) ? creep
   Exit: (15) bubble(charles, [ann, andrew], [charles, ann], andrew)
? creep
   Call: (15) orderofsuccessors([charles, ann], [andrew, edward],
_3820) ? creep
   Call: (16) bubble(charles, [ann], _14602, _14604) ? creep
^  Call: (17) not(successor(charles, ann)) ? creep
   Call: (18) successor(charles, ann) ? creep
   Call: (19) child(elizabeth, charles) ? creep
   Exit: (19) child(elizabeth, charles) ? creep
   Call: (19) child(elizabeth, ann) ? creep
   Exit: (19) child(elizabeth, ann) ? creep
^  Call: (19) not(queen(charles)) ? creep
   Call: (20) queen(charles) ? creep
   Fail: (20) queen(charles) ? creep
^  Exit: (19) not(user:queen(charles)) ? creep
^  Call: (19) not(queen(ann)) ? creep
   Call: (20) queen(ann) ? creep
   Fail: (20) queen(ann) ? creep
^  Exit: (19) not(user:queen(ann)) ? creep
   Call: (19) is_elder(charles, ann) ? creep
   Call: (20) elderSibling(charles, ann) ? creep
   Exit: (20) elderSibling(charles, ann) ? creep
   Exit: (19) is_elder(charles, ann) ? creep
   Exit: (18) successor(charles, ann) ? creep
^  Fail: (17) not(user:successor(charles, ann)) ? creep
   Redo: (16) bubble(charles, [ann], _15564, _15566) ? creep
   Call: (17) successor(charles, ann) ? creep
   Call: (18) child(elizabeth, charles) ? creep
   Exit: (18) child(elizabeth, charles) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
^  Call: (18) not(queen(charles)) ? creep
   Call: (19) queen(charles) ? creep
   Fail: (19) queen(charles) ? creep
^  Exit: (18) not(user:queen(charles)) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
   Call: (18) is_elder(charles, ann) ? creep
   Call: (19) elderSibling(charles, ann) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Exit: (18) is_elder(charles, ann) ? creep
   Exit: (17) successor(charles, ann) ? creep
   Call: (17) bubble(ann, [], _15554, _16428) ? creep
   Exit: (17) bubble(ann, [], [], ann) ? creep
   Exit: (16) bubble(charles, [ann], [charles], ann) ? creep
   Call: (16) orderofsuccessors([charles], [ann, andrew, edward],
_3820) ? creep
   Call: (17) bubble(charles, [], _16608, _16610) ? creep
   Exit: (17) bubble(charles, [], [], charles) ? creep
   Call: (17) orderofsuccessors([], [charles, ann, andrew, edward],
_3820) ? creep
   Exit: (17) orderofsuccessors([], [charles, ann, andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Exit: (16) orderofsuccessors([charles], [ann, andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Exit: (15) orderofsuccessors([charles, ann], [andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Exit: (14) orderofsuccessors([charles, ann, andrew], [edward],
[charles, ann, andrew, edward]) ? creep
   Exit: (13) orderofsuccessors([charles, ann, andrew, edward], [],
[charles, ann, andrew, edward]) ? creep
```

```
   Exit: (12) order_successors([charles, ann, andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Call: (12) write([charles, ann, andrew, edward]) ? creep
[charles,ann,andrew,edward]
   Exit: (12) write([charles, ann, andrew, edward]) ? creep
   Exit: (11) successionList(_3818, [charles, ann, andrew, edward])
? creep
B = [charles, ann, andrew, edward] .
```

To check if Charles, Ann, Andrew, Edward true

```
?- trace, successionList(elizabeth,[charles,ann,andrew,edward]).
   Call: (11) successionList(elizabeth, [charles, ann, andrew,
edward]) ? creep
^  Call: (12) findall(_25702, child(elizabeth, _25702), _25762) ?
creep
   Call: (17) child(elizabeth, _25702) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Redo: (17) child(elizabeth, _25702) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
   Redo: (17) child(elizabeth, _25702) ? creep
   Exit: (17) child(elizabeth, andrew) ? creep
   Redo: (17) child(elizabeth, _25702) ? creep
   Exit: (17) child(elizabeth, edward) ? creep
^  Exit: (12) findall(_25702, user:child(elizabeth, _25702),
[charles, ann, andrew, edward]) ? creep
   Call: (12) order_successors([charles, ann, andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Call: (13) orderofsuccessors([charles, ann, andrew, edward], [],
[charles, ann, andrew, edward]) ? creep
   Call: (14) bubble(charles, [ann, andrew, edward], _26348, _26350)
? creep
^  Call: (15) not(successor(charles, ann)) ? creep
   Call: (16) successor(charles, ann) ? creep
   Call: (17) child(elizabeth, charles) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Call: (17) child(elizabeth, ann) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
^  Call: (17) not(queen(charles)) ? creep
   Call: (18) queen(charles) ? creep
   Fail: (18) queen(charles) ? creep
^  Exit: (17) not(user:queen(charles)) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
   Call: (17) is_elder(charles, ann) ? creep
   Call: (18) elderSibling(charles, ann) ? creep
   Exit: (18) elderSibling(charles, ann) ? creep
   Exit: (17) is_elder(charles, ann) ? creep
   Exit: (16) successor(charles, ann) ? creep
^  Fail: (15) not(user:successor(charles, ann)) ? creep
```

```
   Redo: (14) bubble(charles, [ann, andrew, edward], _27310, _27312)
? creep
   Call: (15) successor(charles, ann) ? creep
   Call: (16) child(elizabeth, charles) ? creep
   Exit: (16) child(elizabeth, charles) ? creep
   Call: (16) child(elizabeth, ann) ? creep
   Exit: (16) child(elizabeth, ann) ? creep
^  Call: (16) not(queen(charles)) ? creep
   Call: (17) queen(charles) ? creep
   Fail: (17) queen(charles) ? creep
^  Exit: (16) not(user:queen(charles)) ? creep
^  Call: (16) not(queen(ann)) ? creep
   Call: (17) queen(ann) ? creep
   Fail: (17) queen(ann) ? creep
^  Exit: (16) not(user:queen(ann)) ? creep
   Call: (16) is_elder(charles, ann) ? creep
   Call: (17) elderSibling(charles, ann) ? creep
   Exit: (17) elderSibling(charles, ann) ? creep
   Exit: (16) is_elder(charles, ann) ? creep
   Exit: (15) successor(charles, ann) ? creep
   Call: (15) bubble(ann, [andrew, edward], _27300, _28174) ? creep
^  Call: (16) not(successor(ann, andrew)) ? creep
   Call: (17) successor(ann, andrew) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
   Call: (18) child(elizabeth, andrew) ? creep
   Exit: (18) child(elizabeth, andrew) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
^  Call: (18) not(queen(andrew)) ? creep
   Call: (19) queen(andrew) ? creep
   Fail: (19) queen(andrew) ? creep
^  Exit: (18) not(user:queen(andrew)) ? creep
   Call: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, andrew) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Exit: (18) is_elder(ann, andrew) ? creep
   Exit: (17) successor(ann, andrew) ? creep
^  Fail: (16) not(user:successor(ann, andrew)) ? creep
   Redo: (15) bubble(ann, [andrew, edward], _27300, _29136) ? creep
   Call: (16) successor(ann, andrew) ? creep
   Call: (17) child(elizabeth, ann) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
   Call: (17) child(elizabeth, andrew) ? creep
   Exit: (17) child(elizabeth, andrew) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
^  Call: (17) not(queen(andrew)) ? creep
   Call: (18) queen(andrew) ? creep
   Fail: (18) queen(andrew) ? creep
```

```
^  Exit: (17) not(user:queen(andrew)) ? creep
   Call: (17) is_elder(ann, andrew) ? creep
   Call: (18) elderSibling(ann, andrew) ? creep
   Exit: (18) elderSibling(ann, andrew) ? creep
   Exit: (17) is_elder(ann, andrew) ? creep
   Exit: (16) successor(ann, andrew) ? creep
   Call: (16) bubble(andrew, [edward], _29124, _29998) ? creep
^  Call: (17) not(successor(andrew, edward)) ? creep
   Call: (18) successor(andrew, edward) ? creep
   Call: (19) child(elizabeth, andrew) ? creep
   Exit: (19) child(elizabeth, andrew) ? creep
   Call: (19) child(elizabeth, edward) ? creep
   Exit: (19) child(elizabeth, edward) ? creep
^  Call: (19) not(queen(andrew)) ? creep
   Call: (20) queen(andrew) ? creep
   Fail: (20) queen(andrew) ? creep
^  Exit: (19) not(user:queen(andrew)) ? creep
^  Call: (19) not(queen(edward)) ? creep
   Call: (20) queen(edward) ? creep
   Fail: (20) queen(edward) ? creep
^  Exit: (19) not(user:queen(edward)) ? creep
   Call: (19) is_elder(andrew, edward) ? creep
   Call: (20) elderSibling(andrew, edward) ? creep
   Exit: (20) elderSibling(andrew, edward) ? creep
   Exit: (19) is_elder(andrew, edward) ? creep
   Exit: (18) successor(andrew, edward) ? creep
^  Fail: (17) not(user:successor(andrew, edward)) ? creep
   Redo: (16) bubble(andrew, [edward], _29124, _30960) ? creep
   Call: (17) successor(andrew, edward) ? creep
   Call: (18) child(elizabeth, andrew) ? creep
   Exit: (18) child(elizabeth, andrew) ? creep
   Call: (18) child(elizabeth, edward) ? creep
   Exit: (18) child(elizabeth, edward) ? creep
^  Call: (18) not(queen(andrew)) ? creep
   Call: (19) queen(andrew) ? creep
   Fail: (19) queen(andrew) ? creep
^  Exit: (18) not(user:queen(andrew)) ? creep
^  Call: (18) not(queen(edward)) ? creep
   Call: (19) queen(edward) ? creep
   Fail: (19) queen(edward) ? creep
^  Exit: (18) not(user:queen(edward)) ? creep
   Call: (18) is_elder(andrew, edward) ? creep
   Call: (19) elderSibling(andrew, edward) ? creep
   Exit: (19) elderSibling(andrew, edward) ? creep
   Exit: (18) is_elder(andrew, edward) ? creep
   Exit: (17) successor(andrew, edward) ? creep
   Call: (17) bubble(edward, [], _30948, _31822) ? creep
   Exit: (17) bubble(edward, [], [], edward) ? creep
   Exit: (16) bubble(andrew, [edward], [andrew], edward) ? creep
   Exit: (15) bubble(ann, [andrew, edward], [ann, andrew], edward) ?
creep
   Exit: (14) bubble(charles, [ann, andrew, edward], [charles, ann,
andrew], edward) ? creep
```

```
   Call: (14) orderofsuccessors([charles, ann, andrew], [edward],
[charles, ann, andrew, edward]) ? creep
   Call: (15) bubble(charles, [ann, andrew], _32090, _32092) ? creep
^  Call: (16) not(successor(charles, ann)) ? creep
   Call: (17) successor(charles, ann) ? creep
   Call: (18) child(elizabeth, charles) ? creep
   Exit: (18) child(elizabeth, charles) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
^  Call: (18) not(queen(charles)) ? creep
   Call: (19) queen(charles) ? creep
   Fail: (19) queen(charles) ? creep
^  Exit: (18) not(user:queen(charles)) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
   Call: (18) is_elder(charles, ann) ? creep
   Call: (19) elderSibling(charles, ann) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Exit: (18) is_elder(charles, ann) ? creep
   Exit: (17) successor(charles, ann) ? creep
^  Fail: (16) not(user:successor(charles, ann)) ? creep
   Redo: (15) bubble(charles, [ann, andrew], _33052, _33054) ? creep
   Call: (16) successor(charles, ann) ? creep
   Call: (17) child(elizabeth, charles) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Call: (17) child(elizabeth, ann) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
^  Call: (17) not(queen(charles)) ? creep
   Call: (18) queen(charles) ? creep
   Fail: (18) queen(charles) ? creep
^  Exit: (17) not(user:queen(charles)) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
   Call: (17) is_elder(charles, ann) ? creep
   Call: (18) elderSibling(charles, ann) ? creep
   Exit: (18) elderSibling(charles, ann) ? creep
   Exit: (17) is_elder(charles, ann) ? creep
   Exit: (16) successor(charles, ann) ? creep
   Call: (16) bubble(ann, [andrew], _33042, _33916) ? creep
^  Call: (17) not(successor(ann, andrew)) ? creep
   Call: (18) successor(ann, andrew) ? creep
   Call: (19) child(elizabeth, ann) ? creep
   Exit: (19) child(elizabeth, ann) ? creep
   Call: (19) child(elizabeth, andrew) ? creep
   Exit: (19) child(elizabeth, andrew) ? creep
^  Call: (19) not(queen(ann)) ? creep
   Call: (20) queen(ann) ? creep
   Fail: (20) queen(ann) ? creep
^  Exit: (19) not(user:queen(ann)) ? creep
^  Call: (19) not(queen(andrew)) ? creep
```

```
   Call: (20) queen(andrew) ? creep
   Fail: (20) queen(andrew) ? creep
^  Exit: (19) not(user:queen(andrew)) ? creep
   Call: (19) is_elder(ann, andrew) ? creep
   Call: (20) elderSibling(ann, andrew) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Exit: (19) is_elder(ann, andrew) ? creep
   Exit: (18) successor(ann, andrew) ? creep
^  Fail: (17) not(user:successor(ann, andrew)) ? creep
   Redo: (16) bubble(ann, [andrew], _33042, _34878) ? creep
   Call: (17) successor(ann, andrew) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
   Call: (18) child(elizabeth, andrew) ? creep
   Exit: (18) child(elizabeth, andrew) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
^  Call: (18) not(queen(andrew)) ? creep
   Call: (19) queen(andrew) ? creep
   Fail: (19) queen(andrew) ? creep
^  Exit: (18) not(user:queen(andrew)) ? creep
   Call: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, andrew) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Exit: (18) is_elder(ann, andrew) ? creep
   Exit: (17) successor(ann, andrew) ? creep
   Call: (17) bubble(andrew, [], _34866, _35740) ? creep
   Exit: (17) bubble(andrew, [], [], andrew) ? creep
   Exit: (16) bubble(ann, [andrew], [ann], andrew) ? creep
   Exit: (15) bubble(charles, [ann, andrew], [charles, ann], andrew)
? creep
   Call: (15) orderofsuccessors([charles, ann], [andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Call: (16) bubble(charles, [ann], _35964, _35966) ? creep
^  Call: (17) not(successor(charles, ann)) ? creep
   Call: (18) successor(charles, ann) ? creep
   Call: (19) child(elizabeth, charles) ? creep
   Exit: (19) child(elizabeth, charles) ? creep
   Call: (19) child(elizabeth, ann) ? creep
   Exit: (19) child(elizabeth, ann) ? creep
^  Call: (19) not(queen(charles)) ? creep
   Call: (20) queen(charles) ? creep
   Fail: (20) queen(charles) ? creep
^  Exit: (19) not(user:queen(charles)) ? creep
^  Call: (19) not(queen(ann)) ? creep
   Call: (20) queen(ann) ? creep
   Fail: (20) queen(ann) ? creep
^  Exit: (19) not(user:queen(ann)) ? creep
   Call: (19) is_elder(charles, ann) ? creep
   Call: (20) elderSibling(charles, ann) ? creep
   Exit: (20) elderSibling(charles, ann) ? creep
   Exit: (19) is_elder(charles, ann) ? creep
   Exit: (18) successor(charles, ann) ? creep
^  Fail: (17) not(user:successor(charles, ann)) ? creep
   Redo: (16) bubble(charles, [ann], _36926, _36928) ? creep
   Call: (17) successor(charles, ann) ? creep
   Call: (18) child(elizabeth, charles) ? creep
   Exit: (18) child(elizabeth, charles) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
^  Call: (18) not(queen(charles)) ? creep
   Call: (19) queen(charles) ? creep
   Fail: (19) queen(charles) ? creep
^  Exit: (18) not(user:queen(charles)) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
   Call: (18) is_elder(charles, ann) ? creep
   Call: (19) elderSibling(charles, ann) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Exit: (18) is_elder(charles, ann) ? creep
   Exit: (17) successor(charles, ann) ? creep
   Call: (17) bubble(ann, [], _36916, _37790) ? creep
   Exit: (17) bubble(ann, [], [], ann) ? creep
   Exit: (16) bubble(charles, [ann], [charles], ann) ? creep
   Call: (16) orderofsuccessors([charles], [ann, andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Call: (17) bubble(charles, [], _37970, _37972) ? creep
   Exit: (17) bubble(charles, [], [], charles) ? creep
   Call: (17) orderofsuccessors([], [charles, ann, andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Exit: (17) orderofsuccessors([], [charles, ann, andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Exit: (16) orderofsuccessors([charles], [ann, andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Exit: (15) orderofsuccessors([charles, ann], [andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Exit: (14) orderofsuccessors([charles, ann, andrew], [edward],
[charles, ann, andrew, edward]) ? creep
   Exit: (13) orderofsuccessors([charles, ann, andrew, edward], [],
[charles, ann, andrew, edward]) ? creep
   Exit: (12) order_successors([charles, ann, andrew, edward],
[charles, ann, andrew, edward]) ? creep
   Call: (12) write([charles, ann, andrew, edward]) ? creep
[charles,ann,andrew,edward]
   Exit: (12) write([charles, ann, andrew, edward]) ? creep
   Exit: (11) successionList(elizabeth, [charles, ann, andrew,
edward]) ? creep
true .
```

To Check if Charles, Andrew, Edward, Ann is the correct succession order

- trace, successionList(elizabeth,[charles,andrew,edward,ann]).

```
   Call: (11) successionList(elizabeth, [charles, andrew, edward,
ann]) ? creep
^  Call: (12) findall(_4344, child(elizabeth, _4344), _4404) ? creep
   Call: (17) child(elizabeth, _4344) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Redo: (17) child(elizabeth, _4344) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
   Redo: (17) child(elizabeth, _4344) ? creep
   Exit: (17) child(elizabeth, andrew) ? creep
   Redo: (17) child(elizabeth, _4344) ? creep
   Exit: (17) child(elizabeth, edward) ? creep
^  Exit: (12) findall(_4344, user:child(elizabeth, _4344), [charles,
ann, andrew, edward]) ? creep
   Call: (12) order_successors([charles, ann, andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Call: (13) orderofsuccessors([charles, ann, andrew, edward], [],
[charles, andrew, edward, ann]) ? creep
   Call: (14) bubble(charles, [ann, andrew, edward], _4990, _4992) ?
creep
^  Call: (15) not(successor(charles, ann)) ? creep
   Call: (16) successor(charles, ann) ? creep
   Call: (17) child(elizabeth, charles) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Call: (17) child(elizabeth, ann) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
^  Call: (17) not(queen(charles)) ? creep
   Call: (18) queen(charles) ? creep
   Fail: (18) queen(charles) ? creep
^  Exit: (17) not(user:queen(charles)) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
   Call: (17) is_elder(charles, ann) ? creep
   Call: (18) elderSibling(charles, ann) ? creep
   Exit: (18) elderSibling(charles, ann) ? creep
   Exit: (17) is_elder(charles, ann) ? creep
   Exit: (16) successor(charles, ann) ? creep
^  Fail: (15) not(user:successor(charles, ann)) ? creep
   Redo: (14) bubble(charles, [ann, andrew, edward], _5952, _5954) ?
creep
   Call: (15) successor(charles, ann) ? creep
   Call: (16) child(elizabeth, charles) ? creep
   Exit: (16) child(elizabeth, charles) ? creep
   Call: (16) child(elizabeth, ann) ? creep
   Exit: (16) child(elizabeth, ann) ? creep
^  Call: (16) not(queen(charles)) ? creep
   Call: (17) queen(charles) ? creep
   Fail: (17) queen(charles) ? creep
^  Exit: (16) not(user:queen(charles)) ? creep
^  Call: (16) not(queen(ann)) ? creep
   Call: (17) queen(ann) ? creep
   Fail: (17) queen(ann) ? creep
^  Exit: (16) not(user:queen(ann)) ? creep
```

```
   Call: (16) is_elder(charles, ann) ? creep
   Call: (17) elderSibling(charles, ann) ? creep
   Exit: (17) elderSibling(charles, ann) ? creep
   Exit: (16) is_elder(charles, ann) ? creep
   Exit: (15) successor(charles, ann) ? creep
   Call: (15) bubble(ann, [andrew, edward], _5942, _6816) ? creep
^  Call: (16) not(successor(ann, andrew)) ? creep
   Call: (17) successor(ann, andrew) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
   Call: (18) child(elizabeth, andrew) ? creep
   Exit: (18) child(elizabeth, andrew) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
^  Call: (18) not(queen(andrew)) ? creep
   Call: (19) queen(andrew) ? creep
   Fail: (19) queen(andrew) ? creep
^  Exit: (18) not(user:queen(andrew)) ? creep
   Call: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, andrew) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Exit: (18) is_elder(ann, andrew) ? creep
   Exit: (17) successor(ann, andrew) ? creep
^  Fail: (16) not(user:successor(ann, andrew)) ? creep
   Redo: (15) bubble(ann, [andrew, edward], _5942, _7778) ? creep
   Call: (16) successor(ann, andrew) ? creep
   Call: (17) child(elizabeth, ann) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
   Call: (17) child(elizabeth, andrew) ? creep
   Exit: (17) child(elizabeth, andrew) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
^  Call: (17) not(queen(andrew)) ? creep
   Call: (18) queen(andrew) ? creep
   Fail: (18) queen(andrew) ? creep
^  Exit: (17) not(user:queen(andrew)) ? creep
   Call: (17) is_elder(ann, andrew) ? creep
   Call: (18) elderSibling(ann, andrew) ? creep
   Exit: (18) elderSibling(ann, andrew) ? creep
   Exit: (17) is_elder(ann, andrew) ? creep
   Exit: (16) successor(ann, andrew) ? creep
   Call: (16) bubble(andrew, [edward], _7766, _8640) ? creep
^  Call: (17) not(successor(andrew, edward)) ? creep
   Call: (18) successor(andrew, edward) ? creep
   Call: (19) child(elizabeth, andrew) ? creep
   Exit: (19) child(elizabeth, andrew) ? creep
   Call: (19) child(elizabeth, edward) ? creep
   Exit: (19) child(elizabeth, edward) ? creep
^  Call: (19) not(queen(andrew)) ? creep
   Call: (20) queen(andrew) ? creep
   Fail: (20) queen(andrew) ? creep
^  Exit: (19) not(user:queen(andrew)) ? creep
^  Call: (19) not(queen(edward)) ? creep
   Call: (20) queen(edward) ? creep
   Fail: (20) queen(edward) ? creep
^  Exit: (19) not(user:queen(edward)) ? creep
   Call: (19) is_elder(andrew, edward) ? creep
   Call: (20) elderSibling(andrew, edward) ? creep
   Exit: (20) elderSibling(andrew, edward) ? creep
   Exit: (19) is_elder(andrew, edward) ? creep
   Exit: (18) successor(andrew, edward) ? creep
^  Fail: (17) not(user:successor(andrew, edward)) ? creep
   Redo: (16) bubble(andrew, [edward], _7766, _9602) ? creep
   Call: (17) successor(andrew, edward) ? creep
   Call: (18) child(elizabeth, andrew) ? creep
   Exit: (18) child(elizabeth, andrew) ? creep
   Call: (18) child(elizabeth, edward) ? creep
   Exit: (18) child(elizabeth, edward) ? creep
^  Call: (18) not(queen(andrew)) ? creep
   Call: (19) queen(andrew) ? creep
   Fail: (19) queen(andrew) ? creep
^  Exit: (18) not(user:queen(andrew)) ? creep
^  Call: (18) not(queen(edward)) ? creep
   Call: (19) queen(edward) ? creep
   Fail: (19) queen(edward) ? creep
^  Exit: (18) not(user:queen(edward)) ? creep
   Call: (18) is_elder(andrew, edward) ? creep
   Call: (19) elderSibling(andrew, edward) ? creep
   Exit: (19) elderSibling(andrew, edward) ? creep
   Exit: (18) is_elder(andrew, edward) ? creep
   Exit: (17) successor(andrew, edward) ? creep
   Call: (17) bubble(edward, [], _9590, _10464) ? creep
   Exit: (17) bubble(edward, [], [], edward) ? creep
   Exit: (16) bubble(andrew, [edward], [andrew], edward) ? creep
   Exit: (15) bubble(ann, [andrew, edward], [ann, andrew], edward) ?
creep
   Exit: (14) bubble(charles, [ann, andrew, edward], [charles, ann,
andrew], edward) ? creep
   Call: (14) orderofsuccessors([charles, ann, andrew], [edward],
[charles, andrew, edward, ann]) ? creep
   Call: (15) bubble(charles, [ann, andrew], _10732, _10734) ? creep
^  Call: (16) not(successor(charles, ann)) ? creep
   Call: (17) successor(charles, ann) ? creep
   Call: (18) child(elizabeth, charles) ? creep
   Exit: (18) child(elizabeth, charles) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
^  Call: (18) not(queen(charles)) ? creep
   Call: (19) queen(charles) ? creep
   Fail: (19) queen(charles) ? creep
^  Exit: (18) not(user:queen(charles)) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
```

```
^  Exit: (18) not(user:queen(ann)) ? creep
   Call: (18) is_elder(charles, ann) ? creep
   Call: (19) elderSibling(charles, ann) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Exit: (18) is_elder(charles, ann) ? creep
   Exit: (17) successor(charles, ann) ? creep
^  Fail: (16) not(user:successor(charles, ann)) ? creep
   Redo: (15) bubble(charles, [ann, andrew], _11694, _11696) ? creep
   Call: (16) successor(charles, ann) ? creep
   Call: (17) child(elizabeth, charles) ? creep
   Exit: (17) child(elizabeth, charles) ? creep
   Call: (17) child(elizabeth, ann) ? creep
   Exit: (17) child(elizabeth, ann) ? creep
^  Call: (17) not(queen(charles)) ? creep
   Call: (18) queen(charles) ? creep
   Fail: (18) queen(charles) ? creep
^  Exit: (17) not(user:queen(charles)) ? creep
^  Call: (17) not(queen(ann)) ? creep
   Call: (18) queen(ann) ? creep
   Fail: (18) queen(ann) ? creep
^  Exit: (17) not(user:queen(ann)) ? creep
   Call: (17) is_elder(charles, ann) ? creep
   Call: (18) elderSibling(charles, ann) ? creep
   Exit: (18) elderSibling(charles, ann) ? creep
   Exit: (17) is_elder(charles, ann) ? creep
   Exit: (16) successor(charles, ann) ? creep
   Call: (16) bubble(ann, [andrew], _11684, _12558) ? creep
^  Call: (17) not(successor(ann, andrew)) ? creep
   Call: (18) successor(ann, andrew) ? creep
   Call: (19) child(elizabeth, ann) ? creep
   Exit: (19) child(elizabeth, ann) ? creep
   Call: (19) child(elizabeth, andrew) ? creep
   Exit: (19) child(elizabeth, andrew) ? creep
^  Call: (19) not(queen(ann)) ? creep
   Call: (20) queen(ann) ? creep
   Fail: (20) queen(ann) ? creep
^  Exit: (19) not(user:queen(ann)) ? creep
^  Call: (19) not(queen(andrew)) ? creep
   Call: (20) queen(andrew) ? creep
   Fail: (20) queen(andrew) ? creep
^  Exit: (19) not(user:queen(andrew)) ? creep
   Call: (19) is_elder(ann, andrew) ? creep
   Call: (20) elderSibling(ann, andrew) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Exit: (19) is_elder(ann, andrew) ? creep
   Exit: (18) successor(ann, andrew) ? creep
^  Fail: (17) not(user:successor(ann, andrew)) ? creep
   Redo: (16) bubble(ann, [andrew], _11684, _13520) ? creep
   Call: (17) successor(ann, andrew) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
   Call: (18) child(elizabeth, andrew) ? creep
   Exit: (18) child(elizabeth, andrew) ? creep
^  Call: (18) not(queen(ann)) ? creep
```

```
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
^  Call: (18) not(queen(andrew)) ? creep
   Call: (19) queen(andrew) ? creep
   Fail: (19) queen(andrew) ? creep
^  Exit: (18) not(user:queen(andrew)) ? creep
   Call: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, andrew) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Exit: (18) is_elder(ann, andrew) ? creep
   Exit: (17) successor(ann, andrew) ? creep
   Call: (17) bubble(andrew, [], _13508, _14382) ? creep
   Exit: (17) bubble(andrew, [], [], andrew) ? creep
   Exit: (16) bubble(ann, [andrew], [ann], andrew) ? creep
   Exit: (15) bubble(charles, [ann, andrew], [charles, ann], andrew)
? creep
   Call: (15) orderofsuccessors([charles, ann], [andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Call: (16) bubble(charles, [ann], _14606, _14608) ? creep
^  Call: (17) not(successor(charles, ann)) ? creep
   Call: (18) successor(charles, ann) ? creep
   Call: (19) child(elizabeth, charles) ? creep
   Exit: (19) child(elizabeth, charles) ? creep
   Call: (19) child(elizabeth, ann) ? creep
   Exit: (19) child(elizabeth, ann) ? creep
^  Call: (19) not(queen(charles)) ? creep
   Call: (20) queen(charles) ? creep
   Fail: (20) queen(charles) ? creep
^  Exit: (19) not(user:queen(charles)) ? creep
^  Call: (19) not(queen(ann)) ? creep
   Call: (20) queen(ann) ? creep
   Fail: (20) queen(ann) ? creep
^  Exit: (19) not(user:queen(ann)) ? creep
   Call: (19) is_elder(charles, ann) ? creep
   Call: (20) elderSibling(charles, ann) ? creep
   Exit: (20) elderSibling(charles, ann) ? creep
   Exit: (19) is_elder(charles, ann) ? creep
   Exit: (18) successor(charles, ann) ? creep
^  Fail: (17) not(user:successor(charles, ann)) ? creep
   Redo: (16) bubble(charles, [ann], _15568, _15570) ? creep
   Call: (17) successor(charles, ann) ? creep
   Call: (18) child(elizabeth, charles) ? creep
   Exit: (18) child(elizabeth, charles) ? creep
   Call: (18) child(elizabeth, ann) ? creep
   Exit: (18) child(elizabeth, ann) ? creep
^  Call: (18) not(queen(charles)) ? creep
   Call: (19) queen(charles) ? creep
   Fail: (19) queen(charles) ? creep
^  Exit: (18) not(user:queen(charles)) ? creep
^  Call: (18) not(queen(ann)) ? creep
   Call: (19) queen(ann) ? creep
   Fail: (19) queen(ann) ? creep
^  Exit: (18) not(user:queen(ann)) ? creep
```

```
   Call: (18) is_elder(charles, ann) ? creep
   Call: (19) elderSibling(charles, ann) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Exit: (18) is_elder(charles, ann) ? creep
   Exit: (17) successor(charles, ann) ? creep
   Call: (17) bubble(ann, [], _15558, _16432) ? creep
   Exit: (17) bubble(ann, [], [], ann) ? creep
   Exit: (16) bubble(charles, [ann], [charles], ann) ? creep
   Call: (16) orderofsuccessors([charles], [ann, andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Call: (17) bubble(charles, [], _16612, _16614) ? creep
   Exit: (17) bubble(charles, [], [], charles) ? creep
   Call: (17) orderofsuccessors([], [charles, ann, andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Fail: (17) orderofsuccessors([], [charles, ann, andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Redo: (17) bubble(charles, [], _16794, _16796) ? creep
   Fail: (17) bubble(charles, [], _16838, _16840) ? creep
   Fail: (16) orderofsuccessors([charles], [ann, andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Redo: (17) bubble(ann, [], _15558, _16928) ? creep
   Fail: (17) bubble(ann, [], _15558, _16972) ? creep
   Redo: (18) is_elder(charles, ann) ? creep
   Call: (19) elderSibling(charles, _17056) ? creep
   Exit: (19) elderSibling(charles, ann) ? creep
   Call: (19) is_elder(ann, ann) ? creep
   Call: (20) elderSibling(ann, ann) ? creep
   Fail: (20) elderSibling(ann, ann) ? creep
   Redo: (19) is_elder(ann, ann) ? creep
   Call: (20) elderSibling(ann, _17320) ? creep
   Exit: (20) elderSibling(ann, andrew) ? creep
   Call: (20) is_elder(andrew, ann) ? creep
   Call: (21) elderSibling(andrew, ann) ? creep
   Fail: (21) elderSibling(andrew, ann) ? creep
   Redo: (20) is_elder(andrew, ann) ? creep
   Call: (21) elderSibling(andrew, _17584) ? creep
   Exit: (21) elderSibling(andrew, edward) ? creep
   Call: (21) is_elder(edward, ann) ? creep
   Call: (22) elderSibling(edward, ann) ? creep
   Fail: (22) elderSibling(edward, ann) ? creep
   Redo: (21) is_elder(edward, ann) ? creep
   Call: (22) elderSibling(edward, _17848) ? creep
   Fail: (22) elderSibling(edward, _17892) ? creep
   Fail: (21) is_elder(edward, ann) ? creep
   Fail: (20) is_elder(andrew, ann) ? creep
   Fail: (19) is_elder(ann, ann) ? creep
   Fail: (18) is_elder(charles, ann) ? creep
   Fail: (17) successor(charles, ann) ? creep
   Fail: (16) bubble(charles, [ann], _18158, _18160) ? creep
   Fail: (15) orderofsuccessors([charles, ann], [andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Redo: (17) bubble(andrew, [], _13508, _18248) ? creep
   Fail: (17) bubble(andrew, [], _13508, _18292) ? creep
   Redo: (18) is_elder(ann, andrew) ? creep
   Call: (19) elderSibling(ann, _18376) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Call: (19) is_elder(andrew, andrew) ? creep
   Call: (20) elderSibling(andrew, andrew) ? creep
   Fail: (20) elderSibling(andrew, andrew) ? creep
   Redo: (19) is_elder(andrew, andrew) ? creep
   Call: (20) elderSibling(andrew, _18640) ? creep
   Exit: (20) elderSibling(andrew, edward) ? creep
   Call: (20) is_elder(edward, andrew) ? creep
   Call: (21) elderSibling(edward, andrew) ? creep
   Fail: (21) elderSibling(edward, andrew) ? creep
   Redo: (20) is_elder(edward, andrew) ? creep
   Call: (21) elderSibling(edward, _18904) ? creep
   Fail: (21) elderSibling(edward, _18948) ? creep
   Fail: (20) is_elder(edward, andrew) ? creep
   Fail: (19) is_elder(andrew, andrew) ? creep
   Fail: (18) is_elder(ann, andrew) ? creep
   Fail: (17) successor(ann, andrew) ? creep
   Fail: (16) bubble(ann, [andrew], _11684, _19172) ? creep
   Redo: (17) is_elder(charles, ann) ? creep
   Call: (18) elderSibling(charles, _19256) ? creep
   Exit: (18) elderSibling(charles, ann) ? creep
   Call: (18) is_elder(ann, ann) ? creep
   Call: (19) elderSibling(ann, ann) ? creep
   Fail: (19) elderSibling(ann, ann) ? creep
   Redo: (18) is_elder(ann, ann) ? creep
   Call: (19) elderSibling(ann, _19520) ? creep
   Exit: (19) elderSibling(ann, andrew) ? creep
   Call: (19) is_elder(andrew, ann) ? creep
   Call: (20) elderSibling(andrew, ann) ? creep
   Fail: (20) elderSibling(andrew, ann) ? creep
   Redo: (19) is_elder(andrew, ann) ? creep
   Call: (20) elderSibling(andrew, _19784) ? creep
   Exit: (20) elderSibling(andrew, edward) ? creep
   Call: (20) is_elder(edward, ann) ? creep
   Call: (21) elderSibling(edward, ann) ? creep
   Fail: (21) elderSibling(edward, ann) ? creep
   Redo: (20) is_elder(edward, ann) ? creep
   Call: (21) elderSibling(edward, _20048) ? creep
   Fail: (21) elderSibling(edward, _20092) ? creep
   Fail: (20) is_elder(edward, ann) ? creep
   Fail: (19) is_elder(andrew, ann) ? creep
   Fail: (18) is_elder(ann, ann) ? creep
   Fail: (17) is_elder(charles, ann) ? creep
   Fail: (16) successor(charles, ann) ? creep
   Fail: (15) bubble(charles, [ann, andrew], _20358, _20360) ? creep
   Fail: (14) orderofsuccessors([charles, ann, andrew], [edward],
[charles, andrew, edward, ann]) ? creep
   Redo: (17) bubble(edward, [], _9590, _20448) ? creep
   Fail: (17) bubble(edward, [], _9590, _20492) ? creep
   Redo: (18) is_elder(andrew, edward) ? creep
   Call: (19) elderSibling(andrew, _20576) ? creep
   Exit: (19) elderSibling(andrew, edward) ? creep
   Call: (19) is_elder(edward, edward) ? creep
```

```
Call: (20) elderSibling(edward, edward) ? creep
Fail: (20) elderSibling(edward, edward) ? creep
Redo: (19) is_elder(edward, edward) ? creep
Call: (20) elderSibling(edward, _20840) ? creep
Fail: (20) elderSibling(edward, _20884) ? creep
Fail: (19) is_elder(edward, edward) ? creep
Fail: (18) is_elder(andrew, edward) ? creep
Fail: (17) successor(andrew, edward) ? creep
Fail: (16) bubble(andrew, [edward], _7766, _21064) ? creep
Redo: (17) is_elder(ann, andrew) ? creep
Call: (18) elderSibling(ann, _21148) ? creep
Exit: (18) elderSibling(ann, andrew) ? creep
Call: (18) is_elder(andrew, andrew) ? creep
Call: (19) elderSibling(andrew, andrew) ? creep
Fail: (19) elderSibling(andrew, andrew) ? creep
Redo: (18) is_elder(andrew, andrew) ? creep
Call: (19) elderSibling(andrew, _21412) ? creep
Exit: (19) elderSibling(andrew, edward) ? creep
Call: (19) is_elder(edward, andrew) ? creep
Call: (20) elderSibling(edward, andrew) ? creep
Fail: (20) elderSibling(edward, andrew) ? creep
Redo: (19) is_elder(edward, andrew) ? creep
Call: (20) elderSibling(edward, _21676) ? creep
Fail: (20) elderSibling(edward, _21720) ? creep
Fail: (19) is_elder(edward, andrew) ? creep
Fail: (18) is_elder(andrew, andrew) ? creep
Fail: (17) is_elder(ann, andrew) ? creep
Fail: (16) successor(ann, andrew) ? creep
Fail: (15) bubble(ann, [andrew, edward], _5942, _21944) ? creep
Redo: (16) is_elder(charles, ann) ? creep
Call: (17) elderSibling(charles, _22028) ? creep
Exit: (17) elderSibling(charles, ann) ? creep
Call: (17) is_elder(ann, ann) ? creep
Call: (18) elderSibling(ann, ann) ? creep
Fail: (18) elderSibling(ann, ann) ? creep
Redo: (17) is_elder(ann, ann) ? creep
Call: (18) elderSibling(ann, _22292) ? creep
Exit: (18) elderSibling(ann, andrew) ? creep
Call: (18) is_elder(andrew, ann) ? creep
Call: (19) elderSibling(andrew, ann) ? creep
Fail: (19) elderSibling(andrew, ann) ? creep
Redo: (18) is_elder(andrew, ann) ? creep
Call: (19) elderSibling(andrew, _22556) ? creep
Exit: (19) elderSibling(andrew, edward) ? creep
Call: (19) is_elder(edward, ann) ? creep
Call: (20) elderSibling(edward, ann) ? creep
Fail: (20) elderSibling(edward, ann) ? creep
Redo: (19) is_elder(edward, ann) ? creep
Call: (20) elderSibling(edward, _22820) ? creep
Fail: (20) elderSibling(edward, _22864) ? creep
Fail: (19) is_elder(edward, ann) ? creep
Fail: (18) is_elder(andrew, ann) ? creep
Fail: (17) is_elder(ann, ann) ? creep
Fail: (16) is_elder(charles, ann) ? creep
```

```
   Fail: (15) successor(charles, ann) ? creep
   Fail: (14) bubble(charles, [ann, andrew, edward], _23130, _23132)
? creep
   Fail: (13) orderofsuccessors([charles, ann, andrew, edward], [],
[charles, andrew, edward, ann]) ? creep
   Fail: (12) order_successors([charles, ann, andrew, edward],
[charles, andrew, edward, ann]) ? creep
   Fail: (11) successionList(elizabeth, [charles, andrew, edward,
ann]) ? creep
false.
```