Exercise 1: The Smart Phone Rivalry

1. **First Order Logic (FOL)**

Competitor(SumSum, Appy)

$\exists x$ Technology(x) $\wedge$ Developed(SumSum, x)

Steal(Stevey, Galactica-S3, SumSum)

Boss(Stevey)

$\forall x,y,z$ Boss(x) ^ Business(y) $\wedge$ Company(z) $\wedge$ Rival(z) $\wedge$ Steal(x,y,z) $\Rightarrow$ Unethical (x)

$\forall x$ Competitor(x, Appy) $\Rightarrow$ Rival(x)

$\forall x$ Technology(x) $\Rightarrow$ Business(x)

Company(SumSum)

Company(Appy)

2. **Prolog clauses**

```
company(sumsum).
company(appy).
competitor(sumsum,appy).
technology(galactica_s3).
developed(sumsum,galactica_s3).
boss(stevey).
business(X) :- technology(X).
rival(X) :- competitor(X,appy).
steal(stevey,galactica_s3,sumsum).
unethical(X) :- boss(X), business(Y), company(Z), rival(Z), steal(X,Y,Z).
```

3. **Trace**

```
?- unethical(stevey).
true .

?- trace.
true.

[trace]  ?- unethical(stevey).
   Call: (8) unethical(stevey) ? creep
   Call: (9) boss(stevey) ? creep
   Exit: (9) boss(stevey) ? creep
   Call: (9) business(_4510) ? creep
   Call: (10) technology(_4510) ? creep
   Exit: (10) technology(galactica_s3) ? creep
   Exit: (9) business(galactica_s3) ? creep
   Call: (9) company(_4510) ? creep
   Exit: (9) company(sumsum) ? creep
   Call: (9) rival(sumsum) ? creep
   Call: (10) competitor(sumsum, appy) ? creep
   Exit: (10) competitor(sumsum, appy) ? creep
   Exit: (9) rival(sumsum) ? creep
   Call: (9) steal(stevey, galactica_s3, sumsum) ? creep
   Exit: (9) steal(stevey, galactica_s3, sumsum) ? creep
   Exit: (8) unethical(stevey) ? creep
true .
```

Exercise 2: The Royal Family

1. **Prolog implementation**

```
female(queenElizabeth).
female(princessAnn).
male(princeCharles).
male(princeAndrew).
male(princeEdward).

parentOf(queenElizabeth, princeCharles).
parentOf(queenElizabeth, princessAnn).
parentOf(queenElizabeth, princeAndrew).
parentOf(queenElizabeth, princeEdward).

olderThan(queenElizabeth, princeCharles).
olderThan(princeCharles, princessAnn).
olderThan(princessAnn, princeAndrew).
olderThan(princeAndrew, princeEdward).

% iteration of the olderThan() cause never permutate all the olderThan() possibilities so need
check if example princeCharles is older than princeEdward
isOlder(X, Y) :- olderThan(X, Y).
isOlder(X, Y) :- olderThan(X, Z), isOlder(Z, Y).

% condition to rank succession
inOrder(X, Y) :- male(X), male(Y), isOlder(X, Y).
inOrder(X, Y) :- male(X), female(Y).
inOrder(X, Y) :- female(X), female(Y), isOlder(X, Y).

% sort in succession order
insertionSort(List, Sorted) :- i_sort(List, [], Sorted).
i_sort([], Acc, Acc).
i_sort([H|T], Acc, Sorted) :- insert(H, Acc, NAcc), i_sort(T, NAcc, Sorted).

insert(X, [Y|T], [Y|NT]) :- inOrder(Y, X), insert(X, T, NT).
insert(X, [Y|T],[X, Y|T]) :- inOrder(X, Y).
insert(X, [], [X]).

% get a list of children using findall() then use insertion sort to sort them in succession order
succession(ListOfSuccession) :- findall(Y, parentOf(queenElizabeth, Y), ListOfChildren),
insertionSort(ListOfChildren, ListOfSuccession).
```

Modifications

Old	New
inOrder(X, Y) :- male(X), male(Y), isOlder(X, Y). inOrder(X, Y) :- male(X), female(Y). inOrder(X, Y) :- female(X), female(Y), isOlder(X, Y).	inOrder(X, Y) :- isOlder(X, Y).

Regarding the modification of the royal succession rule, the rules and prolog knowledge base changed from order of birth of male line before considering female like to only order of birth regardless of the gender. Therefore, the previous conditions in “inOrder(X, Y)” which consider both gender and birth order has been changed to only consider birth order, “isOlder(X, Y)”.

Trace

```
?- trace.
true.

[trace]  ?- succession(X).
   Call: (8) succession(_4330) ? creep
^  Call: (9) findall(_4532, parentOf(queenElizabeth, _4532), _4554) ? creep
   Call: (14) parentOf(queenElizabeth, _4532) ? creep
   Exit: (14) parentOf(queenElizabeth, princeCharles) ? creep
   Redo: (14) parentOf(queenElizabeth, _4532) ? creep
   Exit: (14) parentOf(queenElizabeth, princessAnn) ? creep
   Redo: (14) parentOf(queenElizabeth, _4532) ? creep
   Exit: (14) parentOf(queenElizabeth, princeAndrew) ? creep
   Redo: (14) parentOf(queenElizabeth, _4532) ? creep
   Exit: (14) parentOf(queenElizabeth, princeEdward) ? creep
^  Call: (14) call('$bags':'$destroy_findall_bag') ? creep
^  Exit: (14) call('$bags':'$destroy_findall_bag') ? creep
^  Exit: (9) findall(_4532, user:parentOf(queenElizabeth, _4532), [princeCharles, princessAnn, princeAndrew, princeEdward]) ? creep
   Call: (9) insertionSort([princeCharles, princessAnn, princeAndrew, princeEdward], _4330) ? creep
   Call: (10) i_sort([princeCharles, princessAnn, princeAndrew, princeEdward], [], _4330) ? creep
   Call: (11) insert(princeCharles, [], _4612) ? creep
   Exit: (11) insert(princeCharles, [], [princeCharles]) ? creep
   Call: (11) i_sort([princessAnn, princeAndrew, princeEdward], [princeCharles], _4330) ? creep
   Call: (12) insert(princessAnn, [princeCharles], _4618) ? creep
   Call: (13) inOrder(princeCharles, princessAnn) ? creep
   Call: (14) male(princeCharles) ? creep
   Exit: (14) male(princeCharles) ? creep
   Call: (14) male(princessAnn) ? creep
   Fail: (14) male(princessAnn) ? creep
   Redo: (13) inOrder(princeCharles, princessAnn) ? creep
   Call: (14) male(princeCharles) ? creep
   Exit: (14) male(princeCharles) ? creep
   Call: (14) female(princessAnn) ? creep
   Exit: (14) female(princessAnn) ? creep
   Exit: (13) inOrder(princeCharles, princessAnn) ? creep
   Call: (13) insert(princessAnn, [], _4602) ? creep
   Exit: (13) insert(princessAnn, [], [princessAnn]) ? creep
   Exit: (12) insert(princessAnn, [princeCharles], [princeCharles, princessAnn]) ? creep
   Call: (12) i_sort([princeAndrew, princeEdward], [princeCharles, princessAnn], _4330) ? creep
   Call: (13) insert(princeAndrew, [princeCharles, princessAnn], _4630) ? creep
   Call: (14) inOrder(princeCharles, princeAndrew) ? creep
   Call: (15) male(princeCharles) ? creep
   Exit: (15) male(princeCharles) ? creep
   Call: (15) male(princeAndrew) ? creep
   Exit: (15) male(princeAndrew) ? creep
   Call: (15) isOlder(princeCharles, princeAndrew) ? creep
   Call: (16) olderThan(princeCharles, princeAndrew) ? creep
   Fail: (16) olderThan(princeCharles, princeAndrew) ? creep
   Redo: (15) isOlder(princeCharles, princeAndrew) ? creep
   Call: (16) olderThan(princeCharles, _4634) ? creep
   Exit: (16) olderThan(princeCharles, princessAnn) ? creep
   Call: (16) isOlder(princessAnn, princeAndrew) ? creep
   Call: (17) olderThan(princessAnn, princeAndrew) ? creep
   Exit: (17) olderThan(princessAnn, princeAndrew) ? creep
   Exit: (16) isOlder(princessAnn, princeAndrew) ? creep
   Exit: (15) isOlder(princeCharles, princeAndrew) ? creep
   Exit: (14) inOrder(princeCharles, princeAndrew) ? creep
   Call: (14) insert(princeAndrew, [princessAnn], _4614) ? creep
   Call: (15) inOrder(princessAnn, princeAndrew) ? creep
   Call: (16) male(princessAnn) ? creep
   Fail: (16) male(princessAnn) ? creep
   Redo: (15) inOrder(princessAnn, princeAndrew) ? creep
   Call: (16) male(princessAnn) ? creep
   Fail: (16) male(princessAnn) ? creep
   Redo: (15) inOrder(princessAnn, princeAndrew) ? creep
   Call: (16) female(princessAnn) ? creep
   Exit: (16) female(princessAnn) ? creep
   Call: (16) female(princeAndrew) ? creep
   Fail: (16) female(princeAndrew) ? creep
   Fail: (15) inOrder(princessAnn, princeAndrew) ? creep
   Redo: (14) insert(princeAndrew, [princessAnn], _4614) ? creep
   Call: (15) inOrder(princeAndrew, princessAnn) ? creep
   Call: (16) male(princeAndrew) ? creep
```

```
Exit: (16) male(princeAndrew) ? creep
Call: (16) male(princessAnn) ? creep
Fail: (16) male(princessAnn) ? creep
Redo: (15) inOrder(princeAndrew, princessAnn) ? creep
Call: (16) male(princeAndrew) ? creep
Exit: (16) male(princeAndrew) ? creep
Call: (16) female(princessAnn) ? creep
Exit: (16) female(princessAnn) ? creep
Exit: (15) inOrder(princeAndrew, princessAnn) ? creep
Exit: (14) insert(princeAndrew, [princessAnn], [princeAndrew, princessAnn]) ? creep
Exit: (13) insert(princeAndrew, [princeCharles, princessAnn], [princeCharles, princeAndrew, princessAnn]) ? creep
Call: (13) i_sort([princeEdward], [princeCharles, princeAndrew, princessAnn], _4330) ? creep
Call: (14) insert(princeEdward, [princeCharles, princeAndrew, princessAnn], _4648) ? creep
Call: (15) inOrder(princeCharles, princeEdward) ? creep
Call: (16) male(princeCharles) ? creep
Exit: (16) male(princeCharles) ? creep
Call: (16) male(princeEdward) ? creep
Exit: (16) male(princeEdward) ? creep
Call: (16) isOlder(princeCharles, princeEdward) ? creep
Call: (17) olderThan(princeCharles, princeEdward) ? creep
Fail: (17) olderThan(princeCharles, princeEdward) ? creep
Redo: (16) isOlder(princeCharles, princeEdward) ? creep
Call: (17) olderThan(princeCharles, _4652) ? creep
Exit: (17) olderThan(princeCharles, princessAnn) ? creep
Call: (17) isOlder(princessAnn, princeEdward) ? creep
Call: (18) olderThan(princessAnn, princeEdward) ? creep
Fail: (18) olderThan(princessAnn, princeEdward) ? creep
Redo: (17) isOlder(princessAnn, princeEdward) ? creep
Call: (18) olderThan(princessAnn, _4652) ? creep
Exit: (18) olderThan(princessAnn, princeAndrew) ? creep
Call: (18) isOlder(princeAndrew, princeEdward) ? creep
Call: (19) olderThan(princeAndrew, princeEdward) ? creep
Exit: (19) olderThan(princeAndrew, princeEdward) ? creep
Exit: (18) isOlder(princeAndrew, princeEdward) ? creep
Exit: (17) isOlder(princessAnn, princeEdward) ? creep
Exit: (16) isOlder(princeCharles, princeEdward) ? creep
Exit: (15) inOrder(princeCharles, princeEdward) ? creep
Call: (15) insert(princeEdward, [princeAndrew, princessAnn], _4632) ? creep
Call: (16) inOrder(princeAndrew, princeEdward) ? creep
Call: (17) male(princeAndrew) ? creep
Exit: (17) male(princeAndrew) ? creep
Call: (17) male(princeEdward) ? creep
Exit: (17) male(princeEdward) ? creep
Call: (17) isOlder(princeAndrew, princeEdward) ? creep
Call: (18) olderThan(princeAndrew, princeEdward) ? creep
Exit: (18) olderThan(princeAndrew, princeEdward) ? creep
Exit: (17) isOlder(princeAndrew, princeEdward) ? creep
Exit: (16) inOrder(princeAndrew, princeEdward) ? creep
Call: (16) insert(princeEdward, [princessAnn], _4638) ? creep
Call: (17) inOrder(princessAnn, princeEdward) ? creep
Call: (18) male(princessAnn) ? creep
Fail: (18) male(princessAnn) ? creep
Redo: (17) inOrder(princessAnn, princeEdward) ? creep
Call: (18) male(princessAnn) ? creep
Fail: (18) male(princessAnn) ? creep
Redo: (17) inOrder(princessAnn, princeEdward) ? creep
Call: (18) female(princessAnn) ? creep
Exit: (18) female(princessAnn) ? creep
Call: (18) female(princeEdward) ? creep
Fail: (18) female(princeEdward) ? creep
Fail: (17) inOrder(princessAnn, princeEdward) ? creep
Redo: (16) insert(princeEdward, [princessAnn], _4638) ? creep
Call: (17) inOrder(princeEdward, princessAnn) ? creep
Call: (18) male(princeEdward) ? creep
Exit: (18) male(princeEdward) ? creep
Call: (18) male(princessAnn) ? creep
Fail: (18) male(princessAnn) ? creep
Redo: (17) inOrder(princeEdward, princessAnn) ? creep
Call: (18) male(princeEdward) ? creep
Exit: (18) male(princeEdward) ? creep
Call: (18) female(princessAnn) ? creep
Exit: (18) female(princessAnn) ? creep
Exit: (17) inOrder(princeEdward, princessAnn) ? creep
Exit: (16) insert(princeEdward, [princessAnn], [princeEdward, princessAnn]) ? creep
Exit: (15) insert(princeEdward, [princeAndrew, princessAnn], [princeAndrew, princeEdward, princessAnn]) ? creep
```

```
   Exit: (14) insert(princeEdward, [princeCharles, princeAndrew, princessAnn], [princeCharles, princeAndrew, princeEdward, princessAnn]) ? creep
   Call: (14) i_sort([], [princeCharles, princeAndrew, princeEdward, princessAnn], _4330) ? creep
   Exit: (14) i_sort([], [princeCharles, princeAndrew, princeEdward, princessAnn], [princeCharles, princeAndrew, princeEdward, princessAnn]) ? creep
   Exit: (13) i_sort([princeEdward], [princeCharles, princeAndrew, princessAnn], [princeCharles, princeAndrew, princeEdward, princessAnn]) ? creep
   Exit: (12) i_sort([princeAndrew, princeEdward], [princeCharles, princessAnn], [princeCharles, princeAndrew, princeEdward, princessAnn]) ? creep
   Exit: (11) i_sort([princessAnn, princeAndrew, princeEdward], [princeCharles], [princeCharles, princeAndrew, princeEdward, princessAnn]) ? creep
   Exit: (10) i_sort([princeCharles, princessAnn, princeAndrew, princeEdward], [], [princeCharles, princeAndrew, princeEdward, princessAnn]) ? creep
   Exit: (9) insertionSort([princeCharles, princessAnn, princeAndrew, princeEdward], [princeCharles, princeAndrew, princeEdward, princessAnn]) ? creep
   Exit: (8) succession([princeCharles, princeAndrew, princeEdward, princessAnn]) ? creep
X = [princeCharles, princeAndrew, princeEdward, princessAnn] .
```

2. **Prolog implementation**

```
female(queenElizabeth).
female(princessAnn).
male(princeCharles).
male(princeAndrew).
male(princeEdward).

parentOf(queenElizabeth, princeCharles).
parentOf(queenElizabeth, princessAnn).
parentOf(queenElizabeth, princeAndrew).
parentOf(queenElizabeth, princeEdward).

olderThan(queenElizabeth, princeCharles).
olderThan(princeCharles, princessAnn).
olderThan(princessAnn, princeAndrew).
olderThan(princeAndrew, princeEdward).


% iteration of the olderThan() cause never permutate all the olderThan() possibilities so need
check if example princeCharles is older than princeEdward
isOlder(X, Y) :- olderThan(X, Y).
isOlder(X, Y) :- olderThan(X, Z), isOlder(Z, Y).

% condition to rank succession
inOrder(X, Y) :- isOlder(X, Y).

% sort in succession order based on age due to the new rule
insertionSort(List, Sorted) :- i_sort(List, [], Sorted).
i_sort([], Acc, Acc).
i_sort([H|T], Acc, Sorted) :- insert(H, Acc, NAcc), i_sort(T, NAcc, Sorted).

insert(X, [Y|T], [Y|NT]) :- inOrder(Y, X), insert(X, T, NT).
insert(X, [Y|T],[X, Y|T]) :- inOrder(X, Y).
insert(X, [], [X]).

% get a list of children usng findall() then use insertion sort to sort them in succession order
succession(ListOfSuccession) :- findall(Y, parentOf(queenElizabeth, Y), ListOfChildren),
insertionSort(ListOfChildren, ListOfSuccession).
```

Trace

```
?- trace.
true.

[trace]  ?- succession(X).
   Call: (8) succession(_4330) ? creep
^  Call: (9) findall(_4532, parentOf(queenElizabeth, _4532), _4554) ? creep
   Call: (14) parentOf(queenElizabeth, _4532) ? creep
   Exit: (14) parentOf(queenElizabeth, princeCharles) ? creep
   Redo: (14) parentOf(queenElizabeth, _4532) ? creep
   Exit: (14) parentOf(queenElizabeth, princessAnn) ? creep
   Redo: (14) parentOf(queenElizabeth, _4532) ? creep
   Exit: (14) parentOf(queenElizabeth, princeAndrew) ? creep
   Redo: (14) parentOf(queenElizabeth, _4532) ? creep
   Exit: (14) parentOf(queenElizabeth, princeEdward) ? creep
^  Call: (14) call('$bags':'$destroy_findall_bag') ? creep
^  Exit: (14) call('$bags':'$destroy_findall_bag') ? creep
^  Exit: (9) findall(_4532, user:parentOf(queenElizabeth, _4532), [princeCharles, princessAnn, princeAndrew, princeEdward]) ? creep
   Call: (9) insertionSort([princeCharles, princessAnn, princeAndrew, princeEdward], _4330) ? creep
   Call: (10) i_sort([princeCharles, princessAnn, princeAndrew, princeEdward], [], _4330) ? creep
   Call: (11) insert(princeCharles, [], _4612) ? creep
   Exit: (11) insert(princeCharles, [], [princeCharles]) ? creep
   Call: (11) i_sort([princessAnn, princeAndrew, princeEdward], [princeCharles], _4330) ? creep
   Call: (12) insert(princessAnn, [princeCharles], _4618) ? creep
   Call: (13) inOrder(princeCharles, princessAnn) ? creep
   Call: (14) isOlder(princeCharles, princessAnn) ? creep
   Call: (15) olderThan(princeCharles, princessAnn) ? creep
   Exit: (15) olderThan(princeCharles, princessAnn) ? creep
   Exit: (14) isOlder(princeCharles, princessAnn) ? creep
   Exit: (13) inOrder(princeCharles, princessAnn) ? creep
   Call: (13) insert(princessAnn, [], _4602) ? creep
   Exit: (13) insert(princessAnn, [], [princessAnn]) ? creep
   Exit: (12) insert(princessAnn, [princeCharles], [princeCharles, princessAnn]) ? creep
   Call: (12) i_sort([princeAndrew, princeEdward], [princeCharles, princessAnn], _4330) ? creep
   Call: (13) insert(princeAndrew, [princeCharles, princessAnn], _4630) ? creep
   Call: (14) inOrder(princeCharles, princeAndrew) ? creep
   Call: (15) isOlder(princeCharles, princeAndrew) ? creep
   Call: (16) olderThan(princeCharles, princeAndrew) ? creep
   Fail: (16) olderThan(princeCharles, princeAndrew) ? creep
   Redo: (15) isOlder(princeCharles, princeAndrew) ? creep
   Call: (16) olderThan(princeCharles, _4634) ? creep
   Exit: (16) olderThan(princeCharles, princessAnn) ? creep
   Call: (16) isOlder(princessAnn, princeAndrew) ? creep
   Call: (17) olderThan(princessAnn, princeAndrew) ? creep
   Exit: (17) olderThan(princessAnn, princeAndrew) ? creep
   Exit: (16) isOlder(princessAnn, princeAndrew) ? creep
   Exit: (15) isOlder(princeCharles, princeAndrew) ? creep
   Exit: (14) inOrder(princeCharles, princeAndrew) ? creep
   Call: (14) insert(princeAndrew, [princessAnn], _4614) ? creep
   Call: (15) inOrder(princessAnn, princeAndrew) ? creep
   Call: (16) isOlder(princessAnn, princeAndrew) ? creep
   Call: (17) olderThan(princessAnn, princeAndrew) ? creep
   Exit: (17) olderThan(princessAnn, princeAndrew) ? creep
   Exit: (16) isOlder(princessAnn, princeAndrew) ? creep
   Exit: (15) inOrder(princessAnn, princeAndrew) ? creep
   Call: (15) insert(princeAndrew, [], _4620) ? creep
   Exit: (15) insert(princeAndrew, [], [princeAndrew]) ? creep
   Exit: (14) insert(princeAndrew, [princessAnn], [princessAnn, princeAndrew]) ? creep
   Exit: (13) insert(princeAndrew, [princeCharles, princessAnn], [princeCharles, princessAnn, princeAndrew]) ? creep
   Call: (13) i_sort([princeEdward], [princeCharles, princessAnn, princeAndrew], _4330) ? creep
   Call: (14) insert(princeEdward, [princeCharles, princessAnn, princeAndrew], _4648) ? creep
   Call: (15) inOrder(princeCharles, princeEdward) ? creep
```

```
   Call: (16) isOlder(princeCharles, princeEdward) ? creep
   Call: (17) olderThan(princeCharles, princeEdward) ? creep
   Fail: (17) olderThan(princeCharles, princeEdward) ? creep
   Redo: (16) isOlder(princeCharles, princeEdward) ? creep
   Call: (17) olderThan(princeCharles, _4652) ? creep
   Exit: (17) olderThan(princeCharles, princessAnn) ? creep
   Call: (17) isOlder(princessAnn, princeEdward) ? creep
   Call: (18) olderThan(princessAnn, princeEdward) ? creep
   Fail: (18) olderThan(princessAnn, princeEdward) ? creep
   Redo: (17) isOlder(princessAnn, princeEdward) ? creep
   Call: (18) olderThan(princessAnn, _4652) ? creep
   Exit: (18) olderThan(princessAnn, princeAndrew) ? creep
   Call: (18) isOlder(princeAndrew, princeEdward) ? creep
   Call: (19) olderThan(princeAndrew, princeEdward) ? creep
   Exit: (19) olderThan(princeAndrew, princeEdward) ? creep
   Exit: (18) isOlder(princeAndrew, princeEdward) ? creep
   Exit: (17) isOlder(princessAnn, princeEdward) ? creep
   Exit: (16) isOlder(princeCharles, princeEdward) ? creep
   Exit: (15) inOrder(princeCharles, princeEdward) ? creep
   Call: (15) insert(princeEdward, [princessAnn, princeAndrew], _4632) ? creep
   Call: (16) inOrder(princessAnn, princeEdward) ? creep
   Call: (17) isOlder(princessAnn, princeEdward) ? creep
   Call: (18) olderThan(princessAnn, princeEdward) ? creep
   Fail: (18) olderThan(princessAnn, princeEdward) ? creep
   Redo: (17) isOlder(princessAnn, princeEdward) ? creep
   Call: (18) olderThan(princessAnn, _4658) ? creep
   Exit: (18) olderThan(princessAnn, princeAndrew) ? creep
   Call: (18) isOlder(princeAndrew, princeEdward) ? creep
   Call: (19) olderThan(princeAndrew, princeEdward) ? creep
   Exit: (19) olderThan(princeAndrew, princeEdward) ? creep
   Exit: (18) isOlder(princeAndrew, princeEdward) ? creep
   Exit: (17) isOlder(princessAnn, princeEdward) ? creep
   Exit: (16) inOrder(princessAnn, princeEdward) ? creep
   Call: (16) insert(princeEdward, [princeAndrew], _4638) ? creep
   Call: (17) inOrder(princeAndrew, princeEdward) ? creep
   Call: (18) isOlder(princeAndrew, princeEdward) ? creep
   Call: (19) olderThan(princeAndrew, princeEdward) ? creep
   Exit: (19) olderThan(princeAndrew, princeEdward) ? creep
   Exit: (18) isOlder(princeAndrew, princeEdward) ? creep
   Exit: (17) inOrder(princeAndrew, princeEdward) ? creep
   Call: (17) insert(princeEdward, [], _4644) ? creep
   Exit: (17) insert(princeEdward, [], [princeEdward]) ? creep
   Exit: (16) insert(princeEdward, [princeAndrew], [princeAndrew, princeEdward]) ? creep
   Exit: (15) insert(princeEdward, [princessAnn, princeAndrew], [princessAnn, princeAndrew, princeEdward]) ? creep
   Exit: (14) insert(princeEdward, [princeCharles, princessAnn, princeAndrew], [princeCharles, princessAnn, princeAndrew, princeEdward]) ? creep
   Call: (14) i_sort([], [princeCharles, princessAnn, princeAndrew, princeEdward], _4330) ? creep
   Exit: (14) i_sort([], [princeCharles, princessAnn, princeAndrew, princeEdward], [princeCharles, princessAnn, princeAndrew, princeEdward]) ? creep
   Exit: (13) i_sort([princeEdward], [princeCharles, princessAnn, princeAndrew], [princeCharles, princessAnn, princeAndrew, princeEdward]) ? creep
   Exit: (12) i_sort([princeAndrew, princeEdward], [princeCharles, princessAnn], [princeCharles, princessAnn, princeAndrew, princeEdward]) ? creep
   Exit: (11) i_sort([princessAnn, princeAndrew, princeEdward], [princeCharles], [princeCharles, princessAnn, princeAndrew, princeEdward]) ? creep
   Exit: (10) i_sort([princeCharles, princessAnn, princeAndrew, princeEdward], [], [princeCharles, princessAnn, princeAndrew, princeEdward]) ? creep
   Exit: (9) insertionSort([princeCharles, princessAnn, princeAndrew, princeEdward], [princeCharles, princessAnn, princeAndrew, princeEdward]) ? creep
   Exit: (8) succession([princeCharles, princessAnn, princeAndrew, princeEdward]) ? creep
X = [princeCharles, princessAnn, princeAndrew, princeEdward] .
```