SC3000 Lab Assignment 2

Lab Group: SCS2

Sky Lim En Xing (U2223731A)

Cheah Xue Min Chermine (U2222126F)

Thomas Tan Keat Hao (U2221472J)

Exercise 1: The Smart Phone Rivalry

> sumsum, a competitor of appy, developed some nice smart phone technology called galactica-s3, all of which was stolen by stevey, who is a boss of appy. It is unethical for a boss to steal business from rival companies. A competitor is a rival. Smart phone technology is business.

---

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

Constants/Entities:

- Company(sumsum)
- Company(appy)
- Person(stevey)
- SmartPhoneTechnology(galactica-s3)

Natural language statements to FOL:

> "sumsum, a competitor of appy"

- Competitor(sumsum, appy)

> "sumsum, ... developed some nice smart phone technology called galactica-s3"

- Develop(sumsum, galactica-s3)

> "...galactica-s3, all of which was stolen by stevey"

- Steal(stevey, galactica-s3)

> "stevey, who is a boss of appy"

- Boss(stevey, appy)

> "It is unethical for a boss to steal business from rival companies"

- ∀p ∀a ∀b ∀s Boss(p, a) ∧ Steals(p, s) ∧ Competitor(b, a) ∧ Develop(b, s) ⇒ ¬Ethical(p)

> "A competitor is a rival."

- ∀x ∀y Competitor(x, y) ⇔ Rival(x, y)

> "Smart phone technology is business."

- ∀s SmartPhoneTechnology(s) ⇒ Business(s)

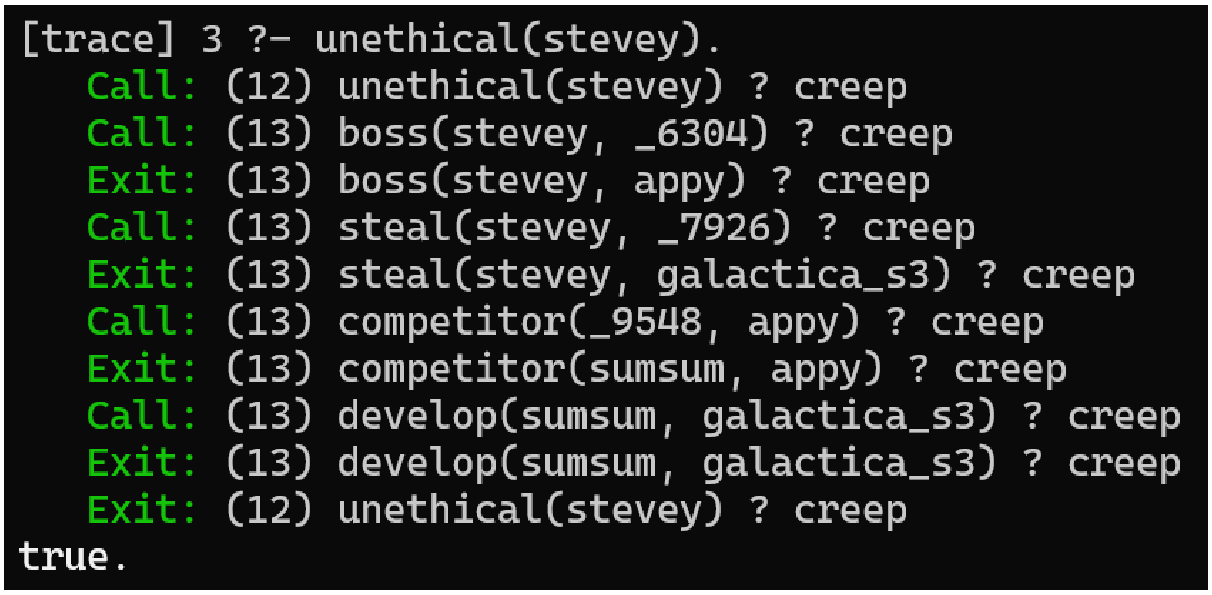
---

2. Write these FOL statements as Prolog clauses.

Refer to attached file 'team6_qn_1_2.pl'.

---

3. Using Prolog, prove that Stevey is unethical. Show a trace of your proof.



```
[trace] 3 ?- unethical(stevey).
   Call: (12) unethical(stevey) ? creep
   Call: (13) boss(stevey, _6304) ? creep
   Exit: (13) boss(stevey, appy) ? creep
   Call: (13) steal(stevey, _7926) ? creep
   Exit: (13) steal(stevey, galactica_s3) ? creep
   Call: (13) competitor(_9548, appy) ? creep
   Exit: (13) competitor(sumsum, appy) ? creep
   Call: (13) develop(sumsum, galactica_s3) ? creep
   Exit: (13) develop(sumsum, galactica_s3) ? creep
   Exit: (12) unethical(stevey) ? creep
true.
```

Exercise 2: The Royal Family

The old Royal succession rule states that the throne is passed down along the male line according to the order of birth before the consideration along the female line – similarly according to the order of birth. queen elizabeth, the monarch of United Kingdom, has four offsprings; namely:- prince charles, princess ann, prince andrew and prince edward – listed in the order of birth.

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

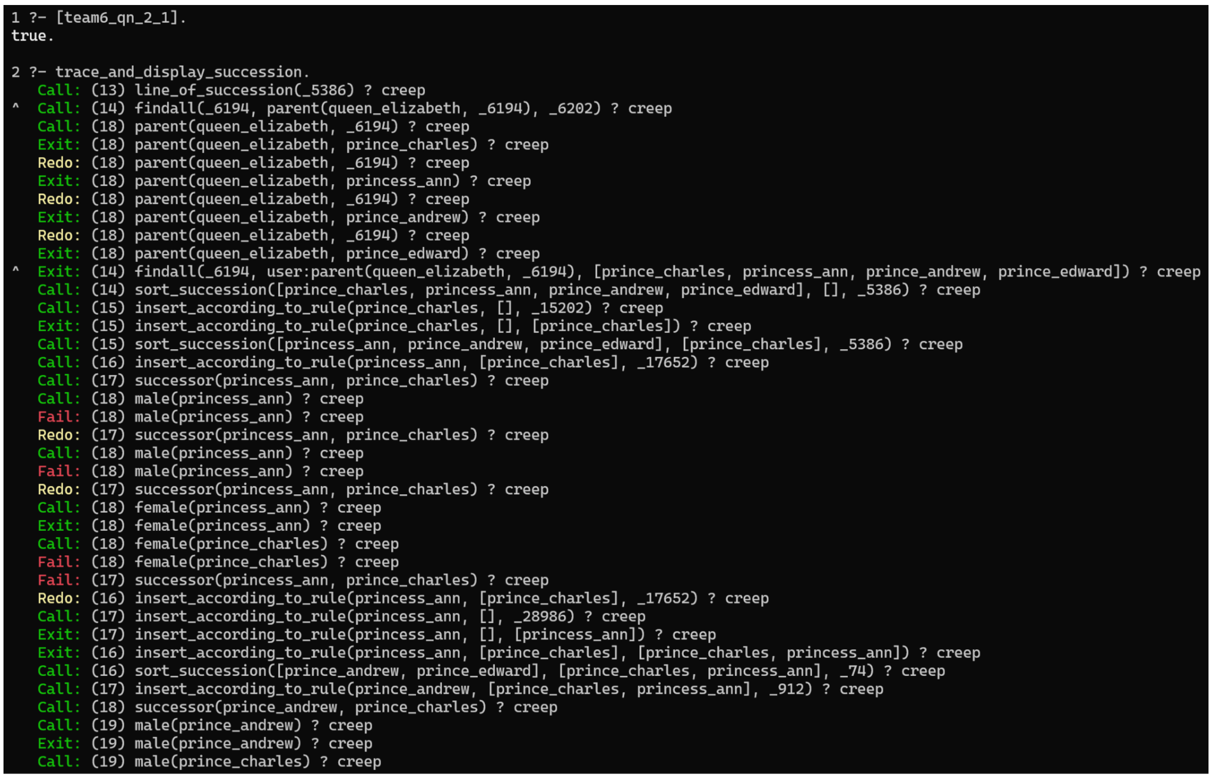
Refer to attached file ' team6_qn_2_1.pl'

Snapshot of Prolog Rule base

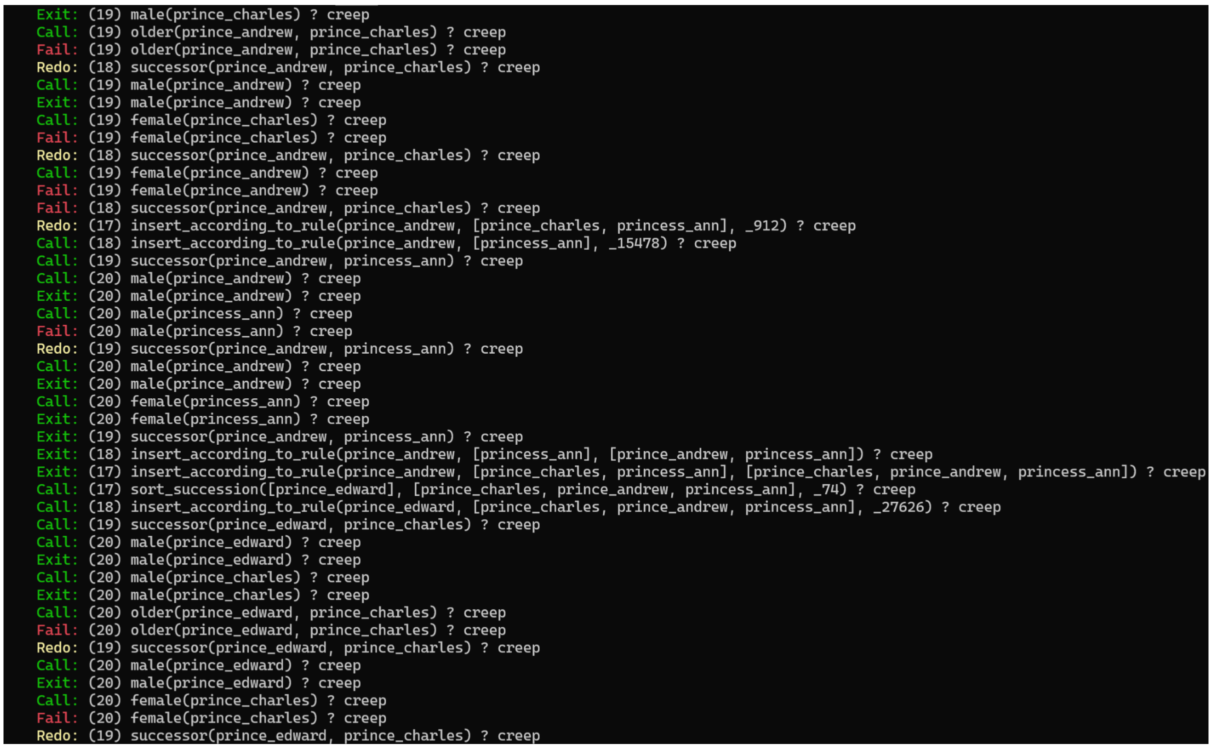
```
% Parent-child relationships
parent(queen_elizabeth, prince_charles).
parent(queen_elizabeth, princess_ann).
parent(queen_elizabeth, prince_andrew).
parent(queen_elizabeth, prince_edward).

% Gender definitions
male(prince_charles).
male(prince_andrew).
male(prince_edward).
female(princess_ann).

% Order of birth
older(prince_charles, princess_ann).
older(princess_ann, prince_andrew).
older(prince_andrew, prince_edward).

% Succession rule: Males come before females, and older siblings come
before younger ones.
successor(X, Y) :- male(X), male(Y), older(X, Y).
successor(X, Y) :- male(X), female(Y).
successor(X, Y) :- female(X), female(Y), older(X, Y).
```

[continue next page]

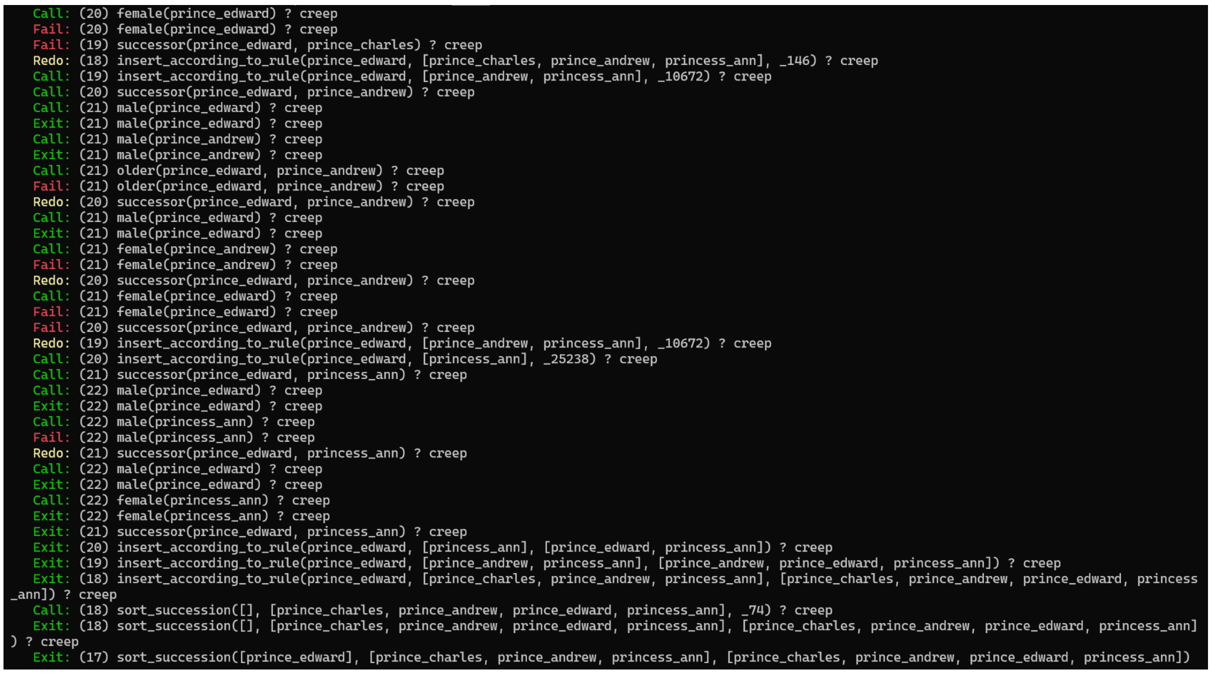
Trace



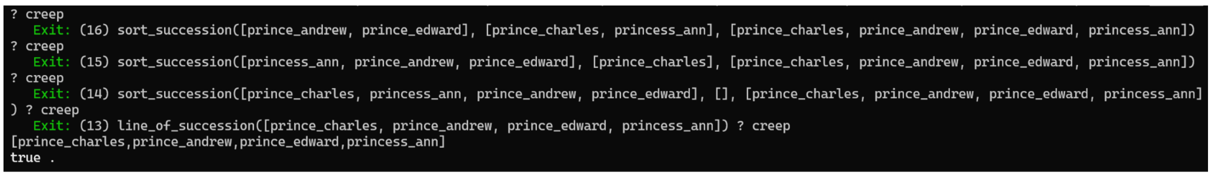
```
1 ?- [team6_qn_2_1].
true.

2 ?- trace_and_display_succession.
   Call: (13) line_of_succession(_5386) ? creep
^  Call: (14) findall(_6194, parent(queen_elizabeth, _6194), _6202) ? creep
   Call: (18) parent(queen_elizabeth, _6194) ? creep
   Exit: (18) parent(queen_elizabeth, prince_charles) ? creep
   Redo: (18) parent(queen_elizabeth, _6194) ? creep
   Exit: (18) parent(queen_elizabeth, princess_ann) ? creep
   Redo: (18) parent(queen_elizabeth, _6194) ? creep
   Exit: (18) parent(queen_elizabeth, prince_andrew) ? creep
   Redo: (18) parent(queen_elizabeth, _6194) ? creep
   Exit: (18) parent(queen_elizabeth, prince_edward) ? creep
^  Exit: (14) findall(_6194, user:parent(queen_elizabeth, _6194), [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
   Call: (14) sort_succession([prince_charles, princess_ann, prince_andrew, prince_edward], [], _5386) ? creep
   Call: (15) insert_according_to_rule(prince_charles, [], _15202) ? creep
   Exit: (15) insert_according_to_rule(prince_charles, [], [prince_charles]) ? creep
   Call: (15) sort_succession([princess_ann, prince_andrew, prince_edward], [prince_charles], _5386) ? creep
   Call: (16) insert_according_to_rule(princess_ann, [prince_charles], _17652) ? creep
   Call: (17) successor(princess_ann, prince_charles) ? creep
   Call: (18) male(princess_ann) ? creep
   Fail: (18) male(princess_ann) ? creep
   Redo: (17) successor(princess_ann, prince_charles) ? creep
   Call: (18) male(princess_ann) ? creep
   Fail: (18) male(princess_ann) ? creep
   Redo: (17) successor(princess_ann, prince_charles) ? creep
   Call: (18) female(princess_ann) ? creep
   Exit: (18) female(princess_ann) ? creep
   Call: (18) female(prince_charles) ? creep
   Fail: (18) female(prince_charles) ? creep
   Fail: (17) successor(princess_ann, prince_charles) ? creep
   Redo: (16) insert_according_to_rule(princess_ann, [prince_charles], _17652) ? creep
   Call: (17) insert_according_to_rule(princess_ann, [], _28986) ? creep
   Exit: (17) insert_according_to_rule(princess_ann, [], [princess_ann]) ? creep
   Exit: (16) insert_according_to_rule(princess_ann, [prince_charles], [prince_charles, princess_ann]) ? creep
   Call: (16) sort_succession([prince_andrew, prince_edward], [prince_charles, princess_ann], _74) ? creep
   Call: (17) insert_according_to_rule(prince_andrew, [prince_charles, princess_ann], _912) ? creep
   Call: (18) successor(prince_andrew, prince_charles) ? creep
   Call: (19) male(prince_andrew) ? creep
   Exit: (19) male(prince_andrew) ? creep
   Call: (19) male(prince_charles) ? creep
```



```
   Exit: (19) male(prince_charles) ? creep
   Call: (19) older(prince_andrew, prince_charles) ? creep
   Fail: (19) older(prince_andrew, prince_charles) ? creep
   Redo: (18) successor(prince_andrew, prince_charles) ? creep
   Call: (19) male(prince_andrew) ? creep
   Exit: (19) male(prince_andrew) ? creep
   Call: (19) female(prince_charles) ? creep
   Fail: (19) female(prince_charles) ? creep
   Redo: (18) successor(prince_andrew, prince_charles) ? creep
   Call: (19) female(prince_andrew) ? creep
   Fail: (19) female(prince_andrew) ? creep
   Fail: (18) successor(prince_andrew, prince_charles) ? creep
   Redo: (17) insert_according_to_rule(prince_andrew, [prince_charles, princess_ann], _912) ? creep
   Call: (18) insert_according_to_rule(prince_andrew, [princess_ann], _15478) ? creep
   Call: (19) successor(prince_andrew, princess_ann) ? creep
   Call: (20) male(prince_andrew) ? creep
   Exit: (20) male(prince_andrew) ? creep
   Call: (20) male(princess_ann) ? creep
   Fail: (20) male(princess_ann) ? creep
   Redo: (19) successor(prince_andrew, princess_ann) ? creep
   Call: (20) male(prince_andrew) ? creep
   Exit: (20) male(prince_andrew) ? creep
   Call: (20) female(princess_ann) ? creep
   Exit: (20) female(princess_ann) ? creep
   Exit: (19) successor(prince_andrew, princess_ann) ? creep
   Exit: (18) insert_according_to_rule(prince_andrew, [princess_ann], [prince_andrew, princess_ann]) ? creep
   Exit: (17) insert_according_to_rule(prince_andrew, [prince_charles, princess_ann], [prince_charles, prince_andrew, princess_ann]) ? creep
   Call: (17) sort_succession([prince_edward], [prince_charles, prince_andrew, princess_ann], _74) ? creep
   Call: (18) insert_according_to_rule(prince_edward, [prince_charles, prince_andrew, princess_ann], _27626) ? creep
   Call: (19) successor(prince_edward, prince_charles) ? creep
   Call: (20) male(prince_edward) ? creep
   Exit: (20) male(prince_edward) ? creep
   Call: (20) male(prince_charles) ? creep
   Exit: (20) male(prince_charles) ? creep
   Call: (20) older(prince_edward, prince_charles) ? creep
   Fail: (20) older(prince_edward, prince_charles) ? creep
   Redo: (19) successor(prince_edward, prince_charles) ? creep
   Call: (20) male(prince_edward) ? creep
   Exit: (20) male(prince_edward) ? creep
   Call: (20) female(prince_charles) ? creep
   Fail: (20) female(prince_charles) ? creep
   Redo: (19) successor(prince_edward, prince_charles) ? creep
```



```
   Call: (20) female(prince_edward) ? creep
   Fail: (20) female(prince_edward) ? creep
   Fail: (19) successor(prince_edward, prince_charles) ? creep
   Redo: (18) insert_according_to_rule(prince_edward, [prince_charles, prince_andrew, princess_ann], _146) ? creep
   Call: (19) insert_according_to_rule(prince_edward, [prince_andrew, princess_ann], _10672) ? creep
   Call: (20) successor(prince_edward, prince_andrew) ? creep
   Call: (21) male(prince_edward) ? creep
   Exit: (21) male(prince_edward) ? creep
   Call: (21) male(prince_andrew) ? creep
   Exit: (21) male(prince_andrew) ? creep
   Call: (21) older(prince_edward, prince_andrew) ? creep
   Fail: (21) older(prince_edward, prince_andrew) ? creep
   Redo: (20) successor(prince_edward, prince_andrew) ? creep
   Call: (21) male(prince_edward) ? creep
   Exit: (21) male(prince_edward) ? creep
   Call: (21) female(prince_andrew) ? creep
   Fail: (21) female(prince_andrew) ? creep
   Redo: (20) successor(prince_edward, prince_andrew) ? creep
   Call: (21) female(prince_edward) ? creep
   Fail: (21) female(prince_edward) ? creep
   Fail: (20) successor(prince_edward, prince_andrew) ? creep
   Redo: (19) insert_according_to_rule(prince_edward, [prince_andrew, princess_ann], _10672) ? creep
   Call: (20) insert_according_to_rule(prince_edward, [princess_ann], _25238) ? creep
   Call: (21) successor(prince_edward, princess_ann) ? creep
   Call: (22) male(prince_edward) ? creep
   Exit: (22) male(prince_edward) ? creep
   Call: (22) male(princess_ann) ? creep
   Fail: (22) male(princess_ann) ? creep
   Redo: (21) successor(prince_edward, princess_ann) ? creep
   Call: (22) male(prince_edward) ? creep
   Exit: (22) male(prince_edward) ? creep
   Call: (22) female(princess_ann) ? creep
   Exit: (22) female(princess_ann) ? creep
   Exit: (21) successor(prince_edward, princess_ann) ? creep
   Exit: (20) insert_according_to_rule(prince_edward, [princess_ann], [prince_edward, princess_ann]) ? creep
   Exit: (19) insert_according_to_rule(prince_edward, [prince_andrew, princess_ann], [prince_andrew, prince_edward, princess_ann]) ? creep
   Exit: (18) insert_according_to_rule(prince_edward, [prince_charles, prince_andrew, princess_ann], [prince_charles, prince_andrew, prince_edward, princess
_ann]) ? creep
   Call: (18) sort_succession([], [prince_charles, prince_andrew, prince_edward, princess_ann], _74) ? creep
   Exit: (18) sort_succession([], [prince_charles, prince_andrew, prince_edward, princess_ann], [prince_charles, prince_andrew, prince_edward, princess_ann]
) ? creep
   Exit: (17) sort_succession([prince_edward], [prince_charles, prince_andrew, princess_ann], [prince_charles, prince_andrew, prince_edward, princess_ann])
```



```
? creep
   Exit: (16) sort_succession([prince_andrew, prince_edward], [prince_charles, princess_ann], [prince_charles, prince_andrew, prince_edward, princess_ann])
? creep
   Exit: (15) sort_succession([princess_ann, prince_andrew, prince_edward], [prince_charles], [prince_charles, prince_andrew, prince_edward, princess_ann])
? creep
   Exit: (14) sort_succession([prince_charles, princess_ann, prince_andrew, prince_edward], [], [prince_charles, prince_andrew, prince_edward, princess_ann]
) ? creep
   Exit: (13) line_of_succession([prince_charles, prince_andrew, prince_edward, princess_ann]) ? creep
[prince_charles,prince_andrew,prince_edward,princess_ann]
true .
```

---

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

Refer to attached file ' team6_qn_2_2.pl'

<u>Snapshot of Prolog Rule base</u>

```
% Parent-child relationships
parent(queen_elizabeth, prince_charles).
parent(queen_elizabeth, princess_ann).
parent(queen_elizabeth, prince_andrew).
parent(queen_elizabeth, prince_edward).

% Gender definitions
male(prince_charles).
male(prince_andrew).
male(prince_edward).
female(princess_ann).
```

```
% Order of birth
older(prince_charles, princess_ann).
older(princess_ann, prince_andrew).
older(prince_andrew, prince_edward).

% New succession rule: Only the order of birth matters.
successor(X, Y) :- older(X, Y).
```

Referring to Exercise 2.1. Snapshot, the only difference in 2.2's Prolog knowledge base and 2.1's is the succession rule, where we now only consider the order of birth, and gender differences are removed from 2.2's knowledge base. Below is a snapshot of the different succession rules, everything else in the code is the same.

Before:

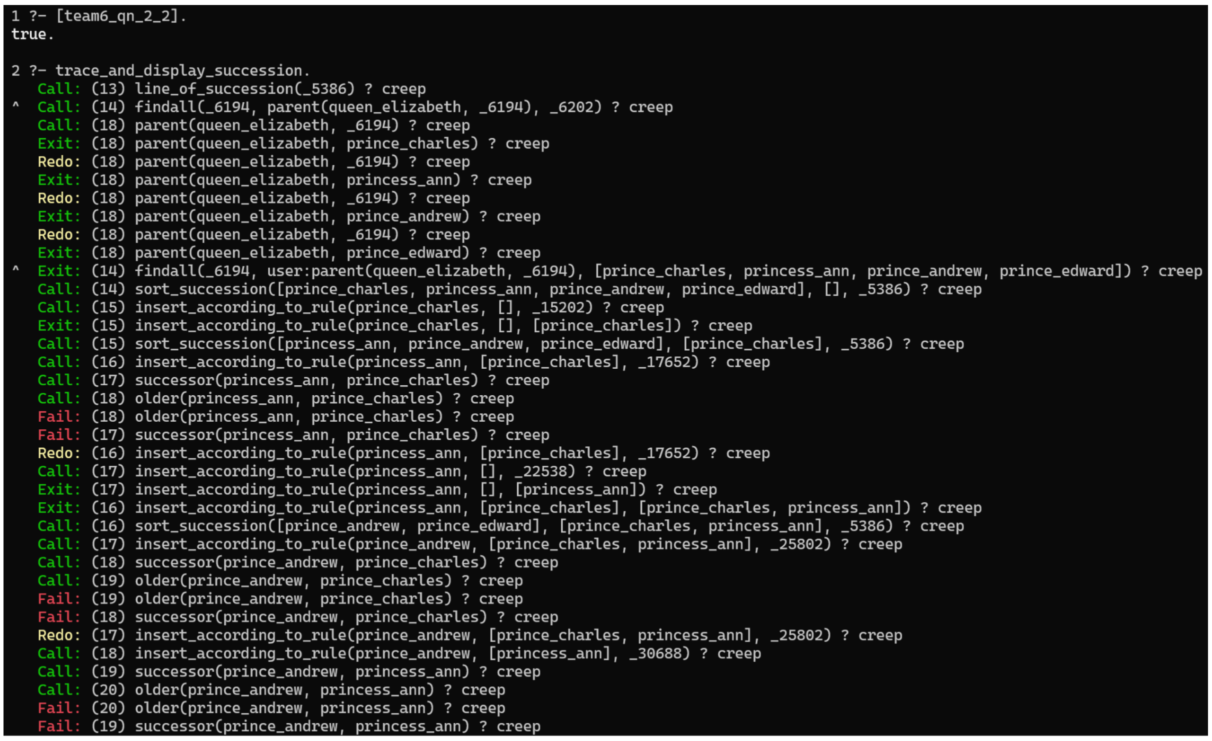
```
% Succession rule: Males come before females, and older siblings come
before younger ones.
successor(X, Y) :- male(X), male(Y), older(X, Y).
successor(X, Y) :- male(X), female(Y).
successor(X, Y) :- female(X), female(Y), older(X, Y).
```

After:

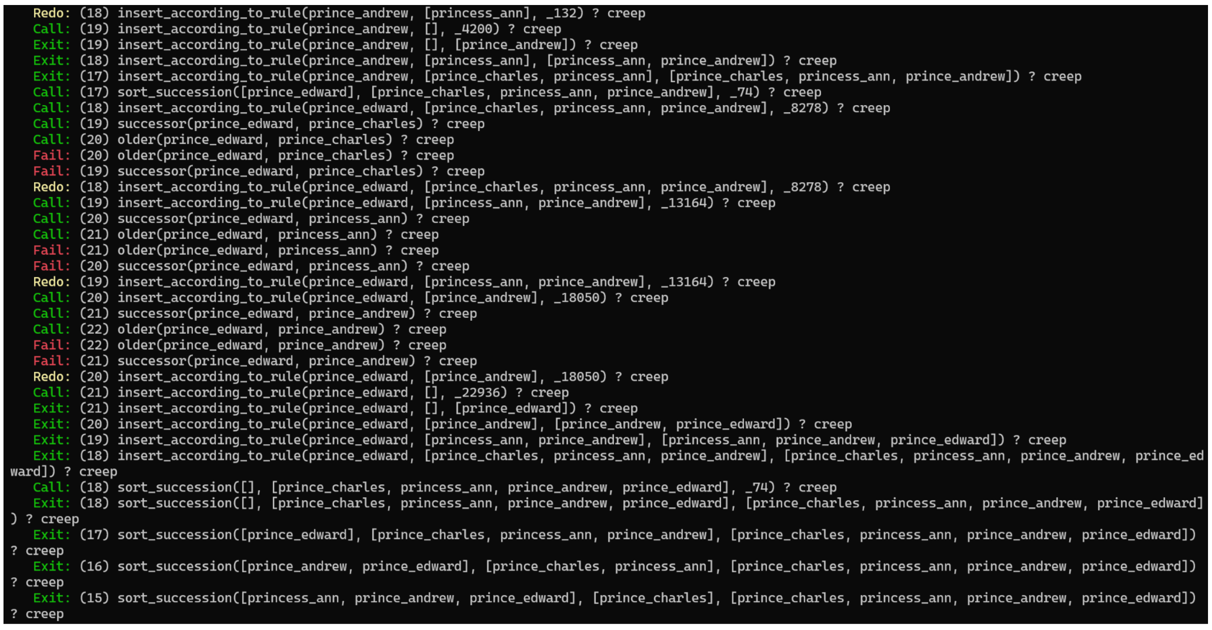
```
% New succession rule: Only the order of birth matters.
successor(X, Y) :- older(X, Y).
```

[continue next page]

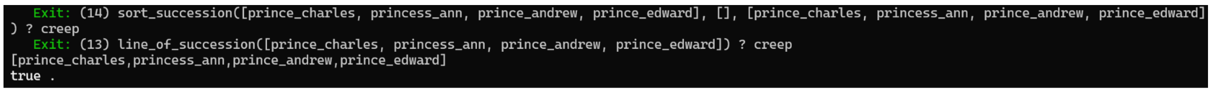
Trace



```
1 ?- [team6_qn_2_2].
true.

2 ?- trace_and_display_succession.
   Call: (13) line_of_succession(_5386) ? creep
^  Call: (14) findall(_6194, parent(queen_elizabeth, _6194), _6202) ? creep
   Call: (18) parent(queen_elizabeth, _6194) ? creep
   Exit: (18) parent(queen_elizabeth, prince_charles) ? creep
   Redo: (18) parent(queen_elizabeth, _6194) ? creep
   Exit: (18) parent(queen_elizabeth, princess_ann) ? creep
   Redo: (18) parent(queen_elizabeth, _6194) ? creep
   Exit: (18) parent(queen_elizabeth, prince_andrew) ? creep
   Redo: (18) parent(queen_elizabeth, _6194) ? creep
   Exit: (18) parent(queen_elizabeth, prince_edward) ? creep
^  Exit: (14) findall(_6194, user:parent(queen_elizabeth, _6194), [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
   Call: (14) sort_succession([prince_charles, princess_ann, prince_andrew, prince_edward], [], _5386) ? creep
   Call: (15) insert_according_to_rule(prince_charles, [], _15202) ? creep
   Exit: (15) insert_according_to_rule(prince_charles, [], [prince_charles]) ? creep
   Call: (15) sort_succession([princess_ann, prince_andrew, prince_edward], [prince_charles], _5386) ? creep
   Call: (16) insert_according_to_rule(princess_ann, [prince_charles], _17652) ? creep
   Call: (17) successor(princess_ann, prince_charles) ? creep
   Call: (18) older(princess_ann, prince_charles) ? creep
   Fail: (18) older(princess_ann, prince_charles) ? creep
   Fail: (17) successor(princess_ann, prince_charles) ? creep
   Redo: (16) insert_according_to_rule(princess_ann, [prince_charles], _17652) ? creep
   Call: (17) insert_according_to_rule(princess_ann, [], _22538) ? creep
   Exit: (17) insert_according_to_rule(princess_ann, [], [princess_ann]) ? creep
   Exit: (16) insert_according_to_rule(princess_ann, [prince_charles], [prince_charles, princess_ann]) ? creep
   Call: (16) sort_succession([prince_andrew, prince_edward], [prince_charles, princess_ann], _5386) ? creep
   Call: (17) insert_according_to_rule(prince_andrew, [prince_charles, princess_ann], _25802) ? creep
   Call: (18) successor(prince_andrew, prince_charles) ? creep
   Call: (19) older(prince_andrew, prince_charles) ? creep
   Fail: (19) older(prince_andrew, prince_charles) ? creep
   Fail: (18) successor(prince_andrew, prince_charles) ? creep
   Redo: (17) insert_according_to_rule(prince_andrew, [prince_charles, princess_ann], _25802) ? creep
   Call: (18) insert_according_to_rule(prince_andrew, [princess_ann], _30688) ? creep
   Call: (19) successor(prince_andrew, princess_ann) ? creep
   Call: (20) older(prince_andrew, princess_ann) ? creep
   Fail: (20) older(prince_andrew, princess_ann) ? creep
   Fail: (19) successor(prince_andrew, princess_ann) ? creep
```



```
   Redo: (18) insert_according_to_rule(prince_andrew, [princess_ann], _132) ? creep
   Call: (19) insert_according_to_rule(prince_andrew, [], _4200) ? creep
   Exit: (19) insert_according_to_rule(prince_andrew, [], [prince_andrew]) ? creep
   Exit: (18) insert_according_to_rule(prince_andrew, [princess_ann], [princess_ann, prince_andrew]) ? creep
   Exit: (17) insert_according_to_rule(prince_andrew, [prince_charles, princess_ann], [prince_charles, princess_ann, prince_andrew]) ? creep
   Call: (17) sort_succession([prince_edward], [prince_charles, princess_ann, prince_andrew], _74) ? creep
   Call: (18) insert_according_to_rule(prince_edward, [prince_charles, princess_ann, prince_andrew], _8278) ? creep
   Call: (19) successor(prince_edward, prince_charles) ? creep
   Call: (20) older(prince_edward, prince_charles) ? creep
   Fail: (20) older(prince_edward, prince_charles) ? creep
   Fail: (19) successor(prince_edward, prince_charles) ? creep
   Redo: (18) insert_according_to_rule(prince_edward, [prince_charles, princess_ann, prince_andrew], _8278) ? creep
   Call: (19) insert_according_to_rule(prince_edward, [princess_ann, prince_andrew], _13164) ? creep
   Call: (20) successor(prince_edward, princess_ann) ? creep
   Call: (21) older(prince_edward, princess_ann) ? creep
   Fail: (21) older(prince_edward, princess_ann) ? creep
   Fail: (20) successor(prince_edward, princess_ann) ? creep
   Redo: (19) insert_according_to_rule(prince_edward, [princess_ann, prince_andrew], _13164) ? creep
   Call: (20) insert_according_to_rule(prince_edward, [prince_andrew], _18050) ? creep
   Call: (21) successor(prince_edward, prince_andrew) ? creep
   Call: (22) older(prince_edward, prince_andrew) ? creep
   Fail: (22) older(prince_edward, prince_andrew) ? creep
   Fail: (21) successor(prince_edward, prince_andrew) ? creep
   Redo: (20) insert_according_to_rule(prince_edward, [prince_andrew], _18050) ? creep
   Call: (21) insert_according_to_rule(prince_edward, [], _22936) ? creep
   Exit: (21) insert_according_to_rule(prince_edward, [], [prince_edward]) ? creep
   Exit: (20) insert_according_to_rule(prince_edward, [prince_andrew], [prince_andrew, prince_edward]) ? creep
   Exit: (19) insert_according_to_rule(prince_edward, [princess_ann, prince_andrew], [princess_ann, prince_andrew, prince_edward]) ? creep
   Exit: (18) insert_according_to_rule(prince_edward, [prince_charles, princess_ann, prince_andrew], [prince_charles, princess_ann, prince_andrew, prince_ed
ward]) ? creep
   Call: (18) sort_succession([], [prince_charles, princess_ann, prince_andrew, prince_edward], _74) ? creep
   Exit: (18) sort_succession([], [prince_charles, princess_ann, prince_andrew, prince_edward], [prince_charles, princess_ann, prince_andrew, prince_edward]
) ? creep
   Exit: (17) sort_succession([prince_edward], [prince_charles, princess_ann, prince_andrew], [prince_charles, princess_ann, prince_andrew, prince_edward])
? creep
   Exit: (16) sort_succession([prince_andrew, prince_edward], [prince_charles, princess_ann], [prince_charles, princess_ann, prince_andrew, prince_edward])
? creep
   Exit: (15) sort_succession([princess_ann, prince_andrew, prince_edward], [prince_charles], [prince_charles, princess_ann, prince_andrew, prince_edward])
? creep
```



```
   Exit: (14) sort_succession([prince_charles, princess_ann, prince_andrew, prince_edward], [], [prince_charles, princess_ann, prince_andrew, prince_edward]
) ? creep
   Exit: (13) line_of_succession([prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
[prince_charles,princess_ann,prince_andrew,prince_edward]
true .
```