



**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  

---

**SINGAPORE**

**School of Computer Science and Engineering**

**CZ3005 - Artificial Intelligence**

**Lab 1 TSP4**

Done By:

Ching Jia Chin

U1620237E

Exercise 1.

1. AppyCompetitor(SumSum)  
OwnerOf(SumSum, Gal-S3)  
SmartPhoneTech(Gal-S3)  
Stole(Stevey, Gal-S3)  
Boss(Stevey)  
Boss & !Stole(Boss, Business) & OwnerOf(Rival, Business) -> Ethical(Boss)  
AppyCompetitor(SumSum) -> Rival(SumSum)  
Business(SmartPhoneTech)

2. `/* Declarations */`  
`appyCompetitor(sumsum).`  
`ownerof(sumsum, gal-s3).`  
`smartphonetech(gal-s3).`  
`stole(stevey, gal-s3).`  
`boss(stevey).`  
`business(smartphonetech).`

`/* functions */`  
`rival(X) :- appyCompetitor(X).`

`ethical(X):-`  
`boss(X),`  
`ownerof(Owner, Business),`  
`rival(Owner),`  
`\+ stole(X, Business).`

- 3.

```
[trace] ?- ethical(stevey).  
Call: (8) ethical(stevey) ? creep  
Call: (9) boss(stevey) ? creep  
Exit: (9) boss(stevey) ? creep  
Call: (9) ownerof(_4686, _4688) ? creep  
Exit: (9) ownerof(sumsum, gal-s3) ? creep  
Call: (9) rival(sumsum) ? creep  
Call: (10) appyCompetitor(sumsum) ? creep  
Exit: (10) appyCompetitor(sumsum) ? creep  
Exit: (9) rival(sumsum) ? creep  
Call: (9) stole(stevey, gal-s3) ? creep  
Exit: (9) stole(stevey, gal-s3) ? creep  
Fail: (8) ethical(stevey) ? creep  
false.
```

## Exercise 2.

1. 

```
/* Declarations*/
offsprings([charles, ann, andrew , edward]). /* Declared according to order of birth*/
male(charles).
male(andrew).
male(edward).
female(ann).

/*sort function for old succession*/
mysort([], [], []).
mysort([H|T], MS, FS):-
    mysort(T, MS0, FS),
        male(H),
    append([H], MS0, MS);
mysort(T, MS, FS0),
female(H),
append([H], FS0, FS).

/*Main Succession call function*/
succession(X) :-
    offsprings(L),
    mysort(L, M, F),
    append(M, F, X).
```

```

[trace] ?- succession(X).
Call: (8) succession(_4500) ? creep
Call: (9) offsprings(_4720) ? creep
Exit: (9) offsprings([charles, ann, andrew, edward]) ? creep
Call: (9) mysort([charles, ann, andrew, edward], _4746, _4748) ? creep
Call: (10) mysort([ann, andrew, edward], _4746, _4748) ? creep
Call: (11) mysort([andrew, edward], _4746, _4748) ? creep
Call: (12) mysort([edward], _4746, _4748) ? creep
Call: (13) mysort([], _4746, _4748) ? creep
Exit: (13) mysort([], [], []) ? creep
Call: (13) male(edward) ? creep
Exit: (13) male(edward) ? creep
Call: (13) lists:append([edward], [], _4754) ? creep
Exit: (13) lists:append([edward], [], [edward]) ? creep
Exit: (12) mysort([edward], [edward], []) ? creep
Call: (12) male(andrew) ? creep
Exit: (12) male(andrew) ? creep
Call: (12) lists:append([andrew], [edward], _4766) ? creep
Exit: (12) lists:append([andrew], [edward], [andrew, edward]) ? creep
Exit: (11) mysort([andrew, edward], [andrew, edward], []) ? creep
Call: (11) male(ann) ? creep
Fail: (11) male(ann) ? creep
Redo: (12) mysort([edward], _4746, _4748) ? creep
Call: (13) mysort([], _4746, _4748) ? creep
Exit: (13) mysort([], [], []) ? creep
Call: (13) female(edward) ? creep
Fail: (13) female(edward) ? creep
Redo: (12) mysort([edward], _4746, _4748) ? creep
Redo: (11) mysort([andrew, edward], _4746, _4748) ? creep
Call: (12) mysort([edward], _4746, _4748) ? creep
Call: (13) mysort([], _4746, _4748) ? creep
Exit: (13) mysort([], [], []) ? creep
Call: (13) male(edward) ? creep
Exit: (13) male(edward) ? creep
Call: (13) lists:append([edward], [], _4754) ? creep
Exit: (13) lists:append([edward], [], [edward]) ? creep
Exit: (12) mysort([edward], [edward], []) ? creep
Call: (12) female(andrew) ? creep
Fail: (12) female(andrew) ? creep
Redo: (12) mysort([edward], _4746, _4748) ? creep
Call: (13) mysort([], _4746, _4748) ? creep
Exit: (13) mysort([], [], []) ? creep
Call: (13) female(edward) ? creep
Fail: (13) female(edward) ? creep
Redo: (12) mysort([edward], _4746, _4748) ? creep
Fail: (11) mysort([andrew, edward], _4746, _4748) ? creep
Redo: (10) mysort([ann, andrew, edward], _4746, _4748) ? creep
Call: (11) mysort([andrew, edward], _4746, _4748) ? creep
Call: (12) mysort([edward], _4746, _4748) ? creep
Call: (13) mysort([], _4746, _4748) ? creep
Exit: (13) mysort([], [], []) ? creep
Call: (13) male(edward) ? creep
Exit: (13) male(edward) ? creep
Call: (13) lists:append([edward], [], _4754) ? creep
Exit: (13) lists:append([edward], [], [edward]) ? creep
Exit: (12) mysort([edward], [edward], []) ? creep
Call: (12) male(andrew) ? creep
Exit: (12) male(andrew) ? creep
Call: (12) lists:append([andrew], [edward], _4766) ? creep
Exit: (12) lists:append([andrew], [edward], [andrew, edward]) ? creep
Exit: (11) mysort([andrew, edward], [andrew, edward], []) ? creep
Call: (11) female(ann) ? creep
Exit: (11) female(ann) ? creep
Call: (11) lists:append([ann], [], _4778) ? creep
Exit: (11) lists:append([ann], [], [ann]) ? creep
Exit: (10) mysort([ann, andrew, edward], [andrew, edward], [ann]) ? creep
Call: (10) male(charles) ? creep
Exit: (10) male(charles) ? creep
Call: (10) lists:append([charles], [andrew, edward], _4790) ? creep
Exit: (10) lists:append([charles], [andrew, edward], [charles, andrew, edward]) ? creep
Exit: (9) mysort([charles, ann, andrew, edward], [charles, andrew, edward], [ann]) ? creep
Call: (9) lists:append([charles, andrew, edward], [ann], _4500) ? creep
Exit: (9) lists:append([charles, andrew, edward], [ann], [charles, andrew, edward, ann]) ? creep
Exit: (8) succession([charles, andrew, edward, ann]) ? creep
X = [charles, andrew, edward, ann] .

```

2. new\_succession(X) :-  
    offsprings(X).

```
[trace] ?- new_succession(X).  
  Call: (8) new_succession(_5412) ? creep  
  Call: (9) offsprings(_5412) ? creep  
  Exit: (9) offsprings([charles, ann, andrew, edward]) ? creep  
  Exit: (8) new_succession([charles, ann, andrew, edward]) ? creep  
X = [charles, ann, andrew, edward].
```