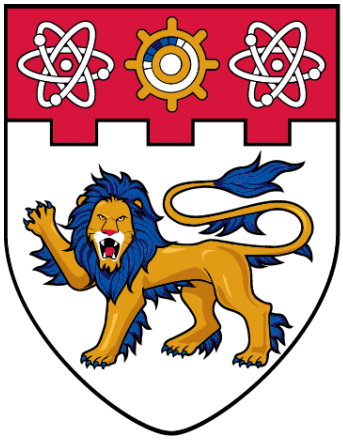


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

# SC3000: Artificial Intelligence

## Assignment 2

## AY24/25 | SEM 2 | SCSB | Group ?


| Name | Matriculation Number |
| --- | --- |
| Ethan Yew Yi Teng | U2321995C |

Qn 1 FOL:

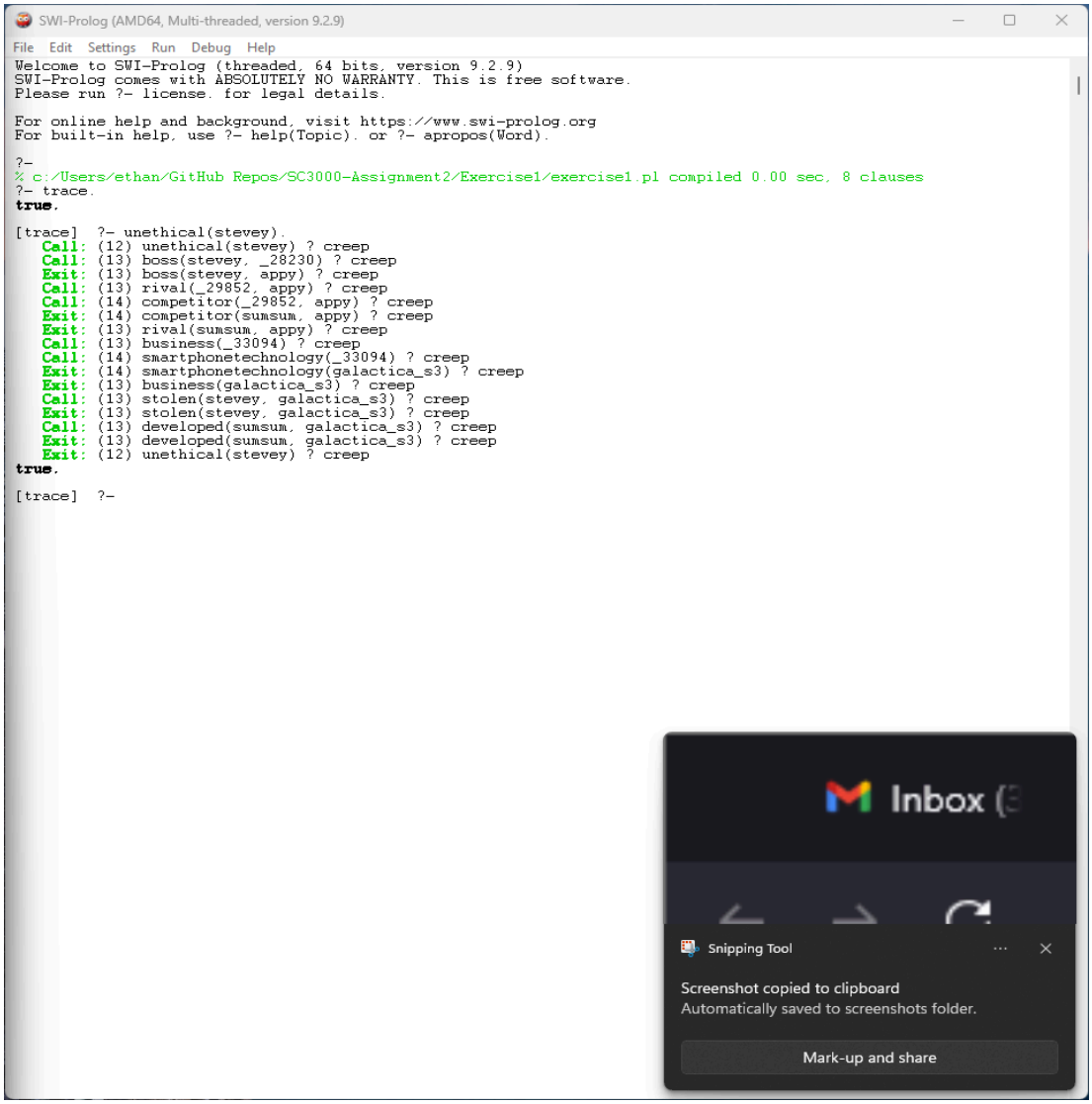
/* First Order Logic */
/*

competitor(sumsum,appy)
developed(sumsum,galactica-s3) ∧ smartphonetechnology(galactica-s3)
stolen(stevey,galactica-s3)
boss(stevey,appy)

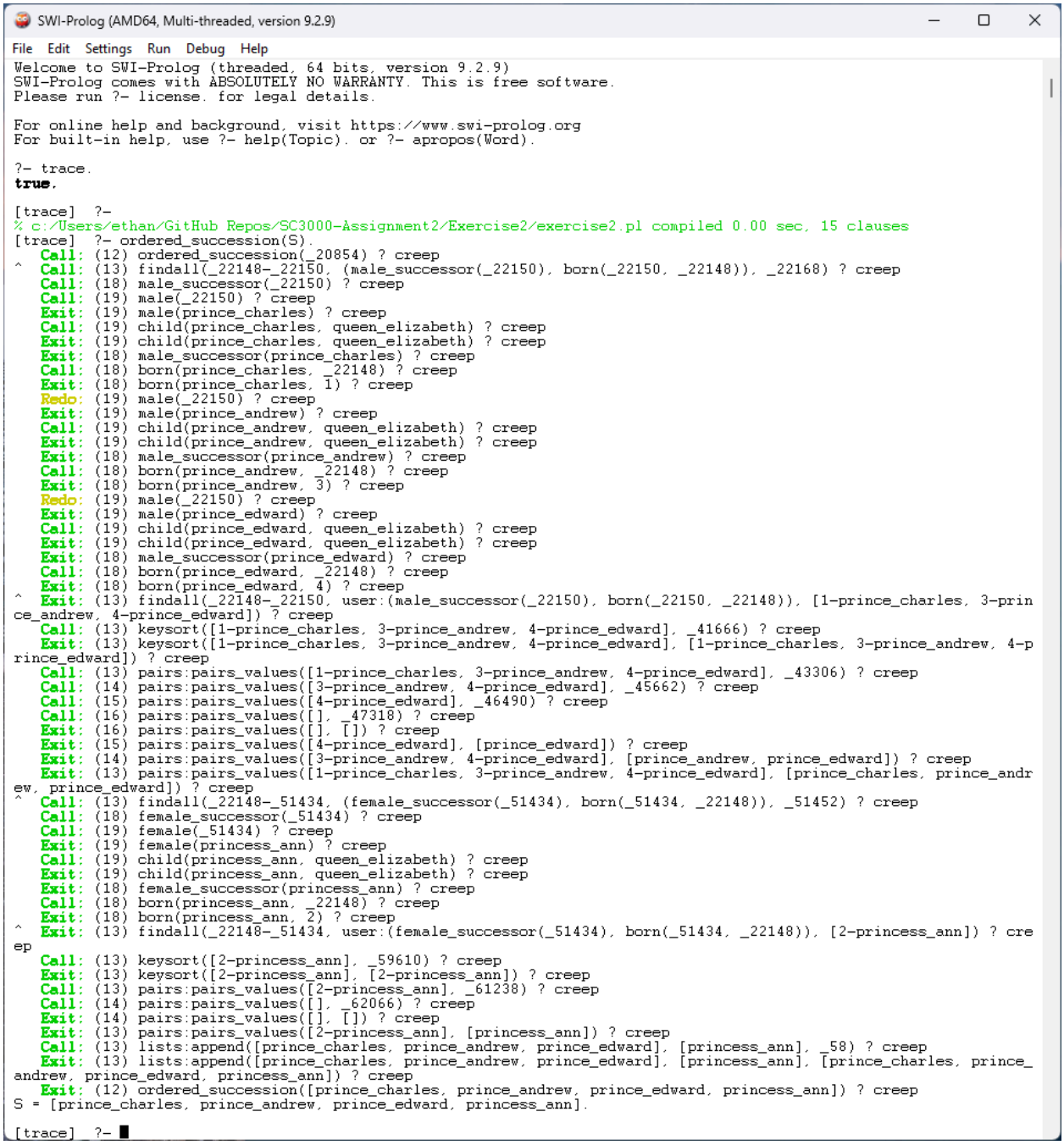
smartphonetechnology(X) -> business(X)
competitor(X,Y) -> rival(X,Y)
boss(X,Y) ∧ rival(Z,Y) ∧ business(W) ∧ stolen(x,w) ∧ developed(z,w) -> unethical(X)

*/

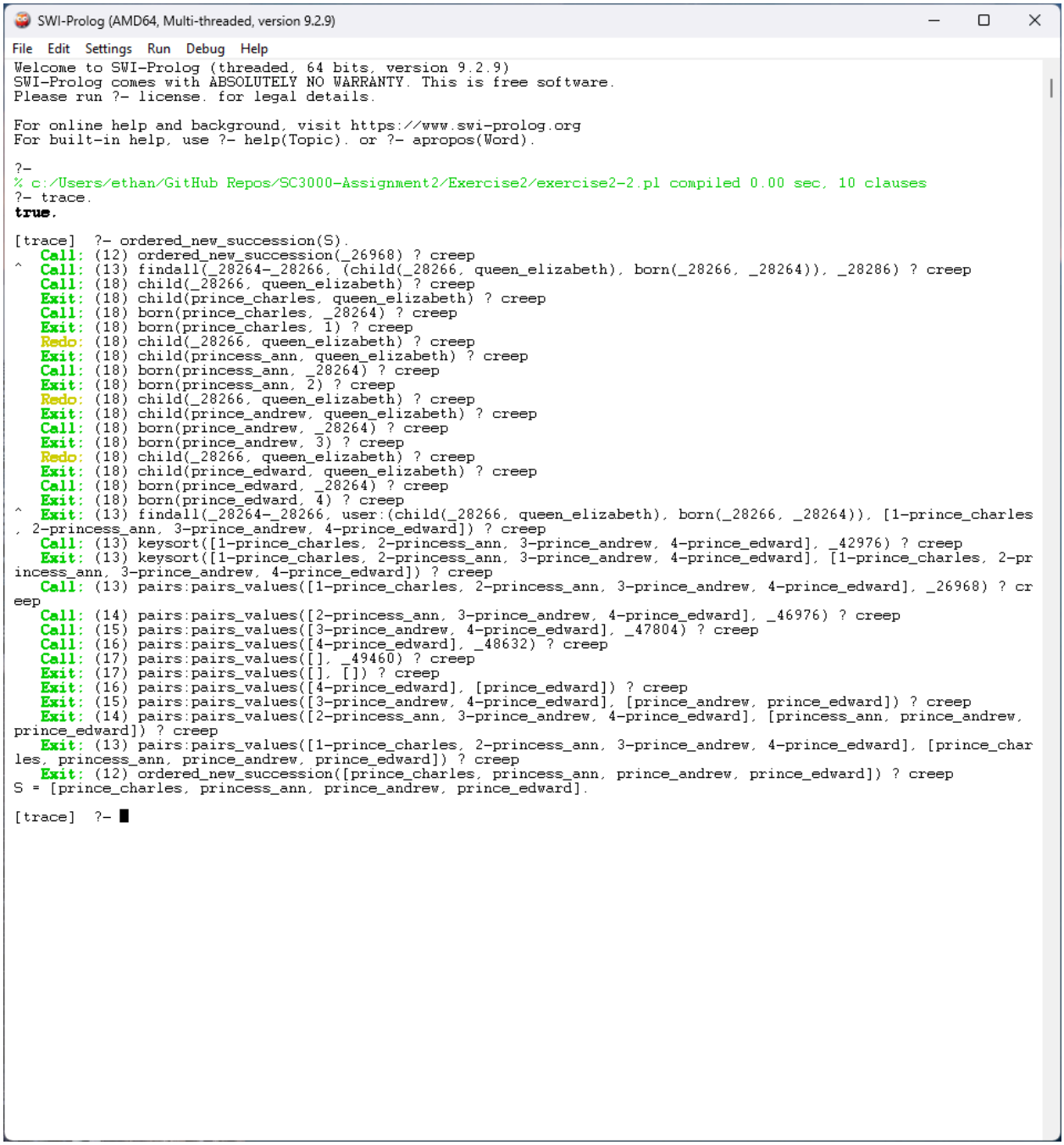
Qn1-2


SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?-
% c:/Users/ethan/GitHub Repos/SC3000-Assignment2/Exercise1/exercise1.pl compiled 0.00 sec, 8 clauses
?- trace.
true.
[trace] ?- unethical(stevey).
Call: (12) unethical(stevey) ? creep
Call: (13) boss(stevey, _28230) ? creep
Exit: (13) boss(stevey, appy) ? creep
Call: (13) rival(_29852, appy) ? creep
Call: (14) competitor(_29852, appy) ? creep
Exit: (14) competitor(sumsum, appy) ? creep
Exit: (13) rival(sumsum, appy) ? creep
Call: (13) business(_33094) ? creep
Call: (14) smartphonetechnology(_33094) ? creep
Exit: (14) smartphonetechnology(galactica_s3) ? creep
Exit: (13) business(galactica_s3) ? creep
Call: (13) stolen(stevey, galactica_s3) ? creep
Exit: (13) stolen(stevey, galactica_s3) ? creep
Call: (13) developed(sumsum, galactica_s3) ? creep
Exit: (13) developed(sumsum, galactica_s3) ? creep
Exit: (12) unethical(stevey) ? creep
true.
[trace] ?-
Inbox
Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Mark-up and share

Q2-1


SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- trace.
true.
[trace] ?-
% c:/Users/ethan/GitHub Repos/SC3000-Assignment2/Exercise2/exercise2.pl compiled 0.00 sec, 15 clauses
[trace] ?- ordered_succession(S).
Call: (12) ordered_succession(_20854) ? creep
^ Call: (13) findall(_22148-_22150, (male_successor(_22150), born(_22150, _22148)), _22168) ? creep
Call: (18) male_successor(_22150) ? creep
Call: (19) male(_22150) ? creep
Exit: (19) male(prince_charles) ? creep
Call: (19) child(prince_charles, queen_elizabeth) ? creep
Exit: (19) child(prince_charles, queen_elizabeth) ? creep
Exit: (18) male_successor(prince_charles) ? creep
Call: (18) born(prince_charles, _22148) ? creep
Exit: (18) born(prince_charles, 1) ? creep
Redo: (19) male(_22150) ? creep
Exit: (19) male(prince_andrew) ? creep
Call: (19) child(prince_andrew, queen_elizabeth) ? creep
Exit: (19) child(prince_andrew, queen_elizabeth) ? creep
Exit: (18) male_successor(prince_andrew) ? creep
Call: (18) born(prince_andrew, _22148) ? creep
Exit: (18) born(prince_andrew, 3) ? creep
Redo: (19) male(_22150) ? creep
Exit: (19) male(prince_edward) ? creep
Call: (19) child(prince_edward, queen_elizabeth) ? creep
Exit: (19) child(prince_edward, queen_elizabeth) ? creep
Exit: (18) male_successor(prince_edward) ? creep
Call: (18) born(prince_edward, _22148) ? creep
Exit: (18) born(prince_edward, 4) ? creep
^ Exit: (13) findall(_22148-_22150, user:(male_successor(_22150), born(_22150, _22148)), [1-prince_charles, 3-prince_andrew, 4-prince_edward]) ? creep
Call: (13) keysort([1-prince_charles, 3-prince_andrew, 4-prince_edward], _41666) ? creep
Exit: (13) keysort([1-prince_charles, 3-prince_andrew, 4-prince_edward], [1-prince_charles, 3-prince_andrew, 4-prince_edward]) ? creep
Call: (13) pairs:pairs_values([1-prince_charles, 3-prince_andrew, 4-prince_edward], _43306) ? creep
Call: (14) pairs:pairs_values([3-prince_andrew, 4-prince_edward], _45662) ? creep
Call: (15) pairs:pairs_values([4-prince_edward], _46490) ? creep
Call: (16) pairs:pairs_values([], _47318) ? creep
Exit: (16) pairs:pairs_values([], []) ? creep
Exit: (15) pairs:pairs_values([4-prince_edward], [prince_edward]) ? creep
Exit: (14) pairs:pairs_values([3-prince_andrew, 4-prince_edward], [prince_andrew, prince_edward]) ? creep
Exit: (13) pairs:pairs_values([1-prince_charles, 3-prince_andrew, 4-prince_edward], [prince_charles, prince_andrew, prince_edward]) ? creep
^ Call: (13) findall(_22148-_51434, (female_successor(_51434), born(_51434, _22148)), _51452) ? creep
Call: (18) female_successor(_51434) ? creep
Call: (19) female(_51434) ? creep
Exit: (19) female(princess_ann) ? creep
Call: (19) child(princess_ann, queen_elizabeth) ? creep
Exit: (19) child(princess_ann, queen_elizabeth) ? creep
Exit: (18) female_successor(princess_ann) ? creep
Call: (18) born(princess_ann, _22148) ? creep
Exit: (18) born(princess_ann, 2) ? creep
^ Exit: (13) findall(_22148-_51434, user:(female_successor(_51434), born(_51434, _22148)), [2-princess_ann]) ? creep
Call: (13) keysort([2-princess_ann], _59610) ? creep
Exit: (13) keysort([2-princess_ann], [2-princess_ann]) ? creep
Call: (13) pairs:pairs_values([2-princess_ann], _61238) ? creep
Call: (14) pairs:pairs_values([], _62066) ? creep
Exit: (14) pairs:pairs_values([], []) ? creep
Exit: (13) pairs:pairs_values([2-princess_ann], [princess_ann]) ? creep
Call: (13) lists:append([prince_charles, prince_andrew, prince_edward], [princess_ann], _58) ? creep
Exit: (13) lists:append([prince_charles, prince_andrew, prince_edward], [princess_ann], [prince_charles, prince_andrew, prince_edward, princess_ann]) ? creep
Exit: (12) ordered_succession([prince_charles, prince_andrew, prince_edward, princess_ann]) ? creep
S = [prince_charles, prince_andrew, prince_edward, princess_ann].
[trace] ?-

Qn2-2


SWI-Prolog (AMD64, Multi-threaded, version 9.2.9)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.9)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?-
% c:/Users/ethan/GitHub Repos/SC3000-Assignment2/Exercise2/exercise2-2.pl compiled 0.00 sec, 10 clauses
?- trace.
true.
[trace] ?- ordered_new_succession(S).
Call: (12) ordered_new_succession(_26968) ? creep
^ Call: (13) findall(_28264-_28266, (child(_28266, queen_elizabeth), born(_28266, _28264)), _28286) ? creep
Call: (18) child(_28266, queen_elizabeth) ? creep
Exit: (18) child(prince_charles, queen_elizabeth) ? creep
Call: (18) born(prince_charles, _28264) ? creep
Exit: (18) born(prince_charles, 1) ? creep
Redo: (18) child(_28266, queen_elizabeth) ? creep
Exit: (18) child(princess_ann, queen_elizabeth) ? creep
Call: (18) born(princess_ann, _28264) ? creep
Exit: (18) born(princess_ann, 2) ? creep
Redo: (18) child(_28266, queen_elizabeth) ? creep
Exit: (18) child(prince_andrew, queen_elizabeth) ? creep
Call: (18) born(prince_andrew, _28264) ? creep
Exit: (18) born(prince_andrew, 3) ? creep
Redo: (18) child(_28266, queen_elizabeth) ? creep
Exit: (18) child(prince_edward, queen_elizabeth) ? creep
Call: (18) born(prince_edward, _28264) ? creep
Exit: (18) born(prince_edward, 4) ? creep
^ Exit: (13) findall(_28264-_28266, user:(child(_28266, queen_elizabeth), born(_28266, _28264)), [1-prince_charles, 2-princess_ann, 3-prince_andrew, 4-prince_edward]) ? creep
Call: (13) keysort([1-prince_charles, 2-princess_ann, 3-prince_andrew, 4-prince_edward], _42976) ? creep
Exit: (13) keysort([1-prince_charles, 2-princess_ann, 3-prince_andrew, 4-prince_edward], [1-prince_charles, 2-princess_ann, 3-prince_andrew, 4-prince_edward]) ? creep
Call: (13) pairs:pairs_values([1-prince_charles, 2-princess_ann, 3-prince_andrew, 4-prince_edward], _26968) ? creep
Call: (14) pairs:pairs_values([2-princess_ann, 3-prince_andrew, 4-prince_edward], _46976) ? creep
Call: (15) pairs:pairs_values([3-prince_andrew, 4-prince_edward], _47804) ? creep
Call: (16) pairs:pairs_values([4-prince_edward], _48632) ? creep
Call: (17) pairs:pairs_values([], _49460) ? creep
Exit: (17) pairs:pairs_values([], []) ? creep
Exit: (16) pairs:pairs_values([4-prince_edward], [prince_edward]) ? creep
Exit: (15) pairs:pairs_values([3-prince_andrew, 4-prince_edward], [prince_andrew, prince_edward]) ? creep
Exit: (14) pairs:pairs_values([2-princess_ann, 3-prince_andrew, 4-prince_edward], [princess_ann, prince_andrew, prince_edward]) ? creep
Exit: (13) pairs:pairs_values([1-prince_charles, 2-princess_ann, 3-prince_andrew, 4-prince_edward], [prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
Exit: (12) ordered_new_succession([prince_charles, princess_ann, prince_andrew, prince_edward]) ? creep
S = [prince_charles, princess_ann, prince_andrew, prince_edward].
[trace] ?-