Question 1

1 (a)

```
owns(waldo, oscar).
owns(sjanie, zappa).
owns(peter, domino).
owns(john, daisy).
owns(andre, penka).

breed(zappa, labrador).
breed(penka, labrador).
breed(oscar, beagle).
breed(domino, dalmation).
breed(daisy, boxer).

service(beagle, hunting).
service(basset, hunting).
service(labrador, guide_dog).
service(german_shepherd, watch_dog).

large(labrador).
medium(beagle).
```

1 (b)

```
PROBLEMS 8   OUTPUT   DEBUG CONSOLE   TERMINAL

Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- ['c:\\users\\chris\\unisa\\2020\\cos4851\\assignments\\assignment01\\question01.pl'].
true.

2 ?- owns(peter, X), breed(X, Y), service(Y, guide_dog).
false.

3 ?- owns(X, Y), breed(Y, german_shepherd).
false.

4 ?- owns(X, domino).
X = peter.

5 ?-
```

1 (c)

0. `owns(X,Y), breed(Y,german_shepherd)` % Initial goal list.
1. `owns(X,Y)` % Scan program for a clause which matches `owns(X,Y)`

2. `owns(waldo,oscar)` % Found `owns(waldo,oscar)` so instantiate `X=waldo, Y=oscar`, goal list becomes `breed(oscar,german_shepherd)`
3. `breed(oscar,german_shepherd)` % Scan program for a clause which matches `breed(oscar,german_shepherd)`.
4. `breed(oscar,german_shepherd)` % No matches found, backtrack and goto step 5.
5. `owns(X,Y)` % Scan program for a clause which matches `owns(X,Y)`
6. `owns(sjanie,zappa)` % Found `owns(sjanie,zappa)` so instantiate `X=sjanie, Y=zappa`, goal list becomes `breed(zappa,german_shepherd)`
7. `breed(zappa,german_shepherd)` % Scan program for a clause which matches `breed(zappa,german_shepherd)`.
8. `breed(zappa,german_shepherd)` % No matches found, backtrack and goto step 9.
9. `owns(X,Y)` % Scan program for a clause which matches `owns(X,Y)`
10. `owns(peter,domino)` % Found `owns(peter,domino)` so instantiate `X=peter, Y=domino`, goal list becomes `breed(domino,german_shepherd)`
11. `breed(domino,german_shepherd)` % Scan program for a clause which matches `breed(domino,german_shepherd)`.
12. `breed(domino,german_shepherd)` % No matches found, backtrack and goto step 13.
13. `owns(X,Y)` % Scan program for a clause which matches `owns(X,Y)`
14. `owns(john,daisy)` % Found `owns(john,daisy)` so instantiate `X=john, Y=daisy`, goal list becomes `breed(daisy,german_shepherd)`
15. `breed(daisy,german_shepherd)` % Scan program for a clause which matches `breed(daisy,german_shepherd)`.
16. `breed(daisy,german_shepherd)` % No matches found, backtrack and goto step 17.
17. `owns(X,Y)` % Scan the program for a clause which matches `owns(X,Y)`
18. `owns(andre,penka)` % Found `owns(andre,penka)` so instantiate `X=andre, Y=penka`, goal list becomes `breed(penka,german_shepherd)`
19. `breed(penka,german_shepherd)` % Scan program for a clause which matches `breed(penka,german_shepherd)`.
20. `breed(penka,german_shepherd)` % No matches found, backtrack and goto step 21.
21. There are no more goals to match against `owns(X,Y)` so there can be no solution. Return false

1 (d)

```
count_breeds(C) :-
    aggregate_all(set(X), breed(_, X), L), % create a unique set of all the breeds
    length(L, C). % assign the length of the set to the variable C.
```

```
PROBLEMS 9  OUTPUT  DEBUG CONSOLE  TERMINAL

Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- ['c:\\Users\\chris\\unisa\\2020\\COS4851\\Assignments\\assignment01\\question01.pl'].
true.

2 ?- count_breeds(X).
X = 4.

3 ?-
```

Question 2

```
exp(Base, Exponent, Result) :-
    % Base case 1 is any positive integer raised to the power 0 should return 1.
    Exponent =:= 0,
    Base >= 0,
    Result is 1
    ;
    % Base case 2 is 0 raised to any positive power should return 0.
    Base =:= 0,
    Exponent > 0, % raising to 0 is handled in base case 1.
    Result is 0
    ;
    % all other positive bases and exponents get handled recursively.
    Base > 0,
    Exponent > 0,
    NewExponent is Exponent - 1, % recursively calculate smaller powers of the base.
    exp(Base, NewExponent, RecursiveResult),
    Result is Base * RecursiveResult.
```

```
PROBLEMS 10  OUTPUT  DEBUG CONSOLE  TERMINAL

Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- ['c:\\Users\\chris\\unisa\\2020\\COS4851\\Assignments\\assignment01\\question02.pl'].
true.

2 ?- exp(2,3,R).
R = 8 ;
false.

3 ?-
```

Question 3

3 (a)

```
student('Keira Clancy', 'female', 4, []).
student('Chris Steenkamp', 'male', 12, ['soccer']).
student('David Steenkamp', 'male', 9, ['cricket']).
student('Jayson Clancy', 'male', 7, ['cricket', 'hockey']).
```

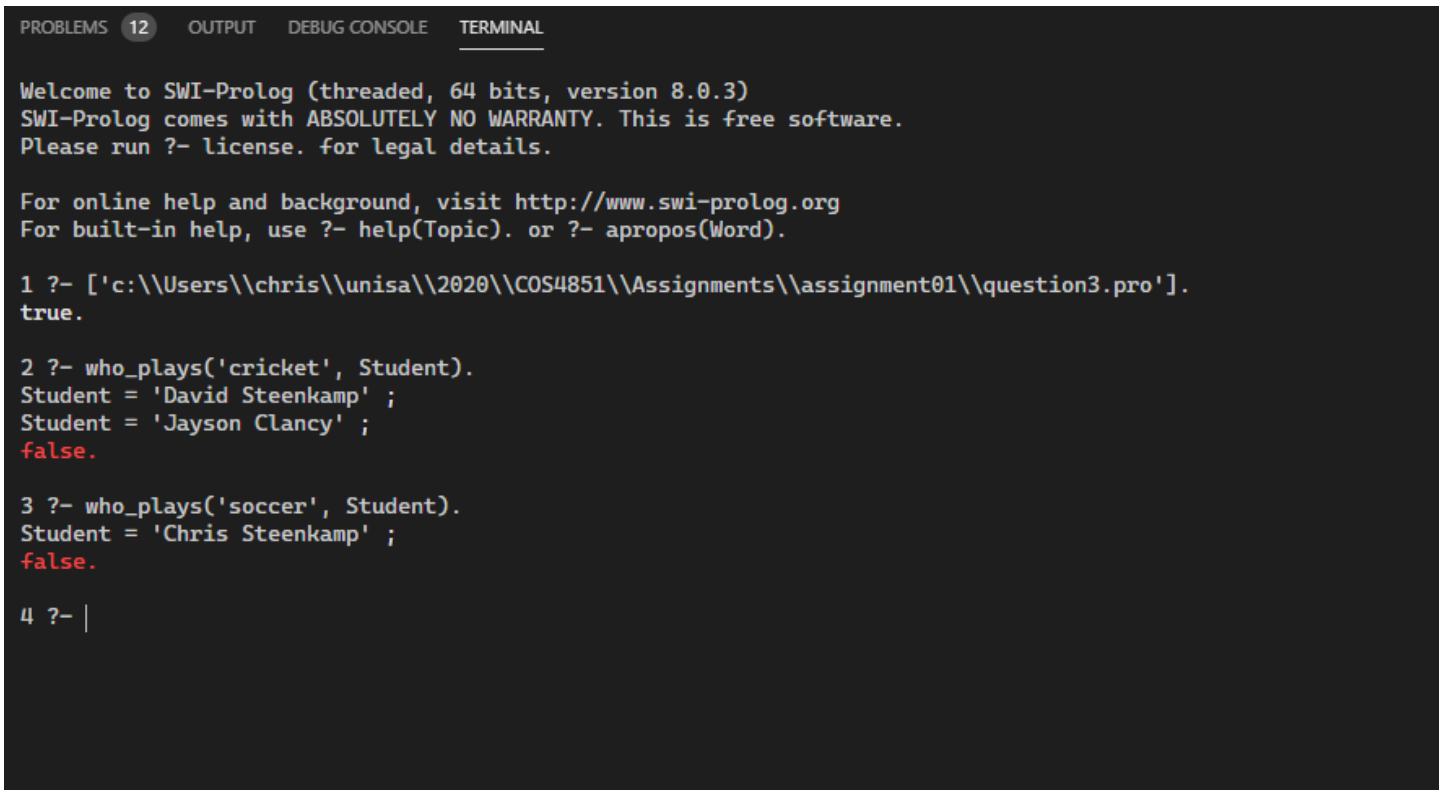
3 (b)

```
gender(Name, Gender) :-
    student(Name, Gender, _, _).

grade(Name, Grade) :-
    student(Name, _, Grade, _).

sports(Name, Sports) :-
    student(Name, _, _, Sports).
```

```
PROBLEMS 9  OUTPUT  DEBUG CONSOLE  TERMINAL

Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- ['c:\\Users\\chris\\unisa\\2020\\COS4851\\Assignments\\assignment01\\question3.pro'].
true.

2 ?- gender('Chris Steenkamp', X).
X = male.

3 ?- gender('Keira Clancy', X).
X = female.

4 ?- grade('David Steenkamp', X).
X = 9.

5 ?- grade('Jayson Clancy', X).
X = 7.

6 ?- sports('Keira Clancy', X).
X = [].

7 ?- sports('Jayson Clancy', X).
X = [cricket, hockey].

8 ?-
```

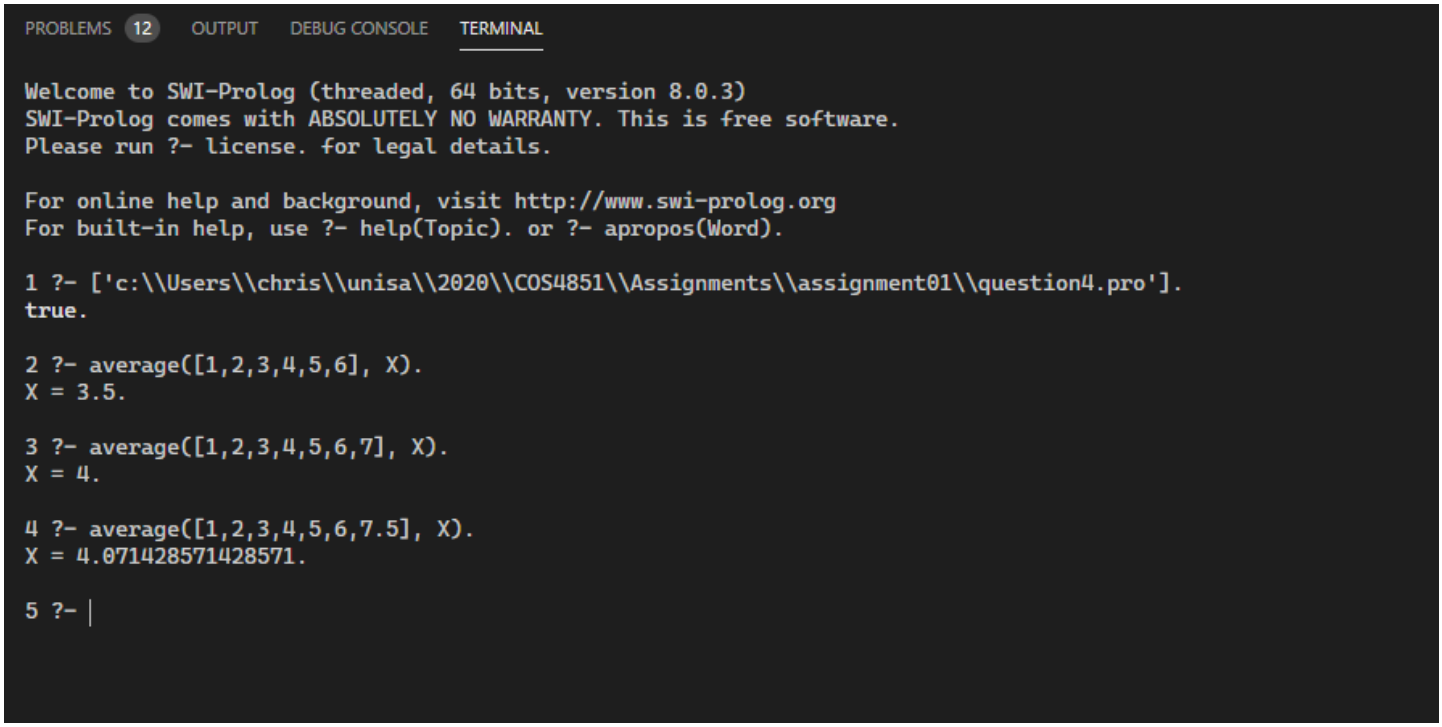
3 (c)

```
member(X, [X|_]).
member(X, [_|Tail]) :-
    member(X, Tail).

who_plays(Sport, Student) :-
    student(Student, _, _, List),
    member(Sport, List).
```



Question 4

```
get_length([], 0).

get_length([_|Tail], Sum) :-
    get_length(Tail, L2),
    Sum is 1 + L2.

sum_list([], 0).

sum_list([X|Tail], Sum) :-
    sum_list(Tail, RecSum),
    Sum is X + RecSum.

average(List, Avg) :-
    sum_list(List, Sum),
    get_length(List, Length),
    Avg is Sum / Length.
```



Question 5

5 (a)

Yes, `[[a, b, c, d, e]]` and `[X | Y]` match.
`X` can be instantiated to `[a,b,c,d,e]` and `Y` can be instantiated to `[]`.

5 (b)

No, `[a, b, c, d, e]` and `[[ X ]|[ b, c, d, e ]]` do not match.
There is no way that the first item in the list, which is the single atom `a`, can be instantiated to match a list.

5 (c)

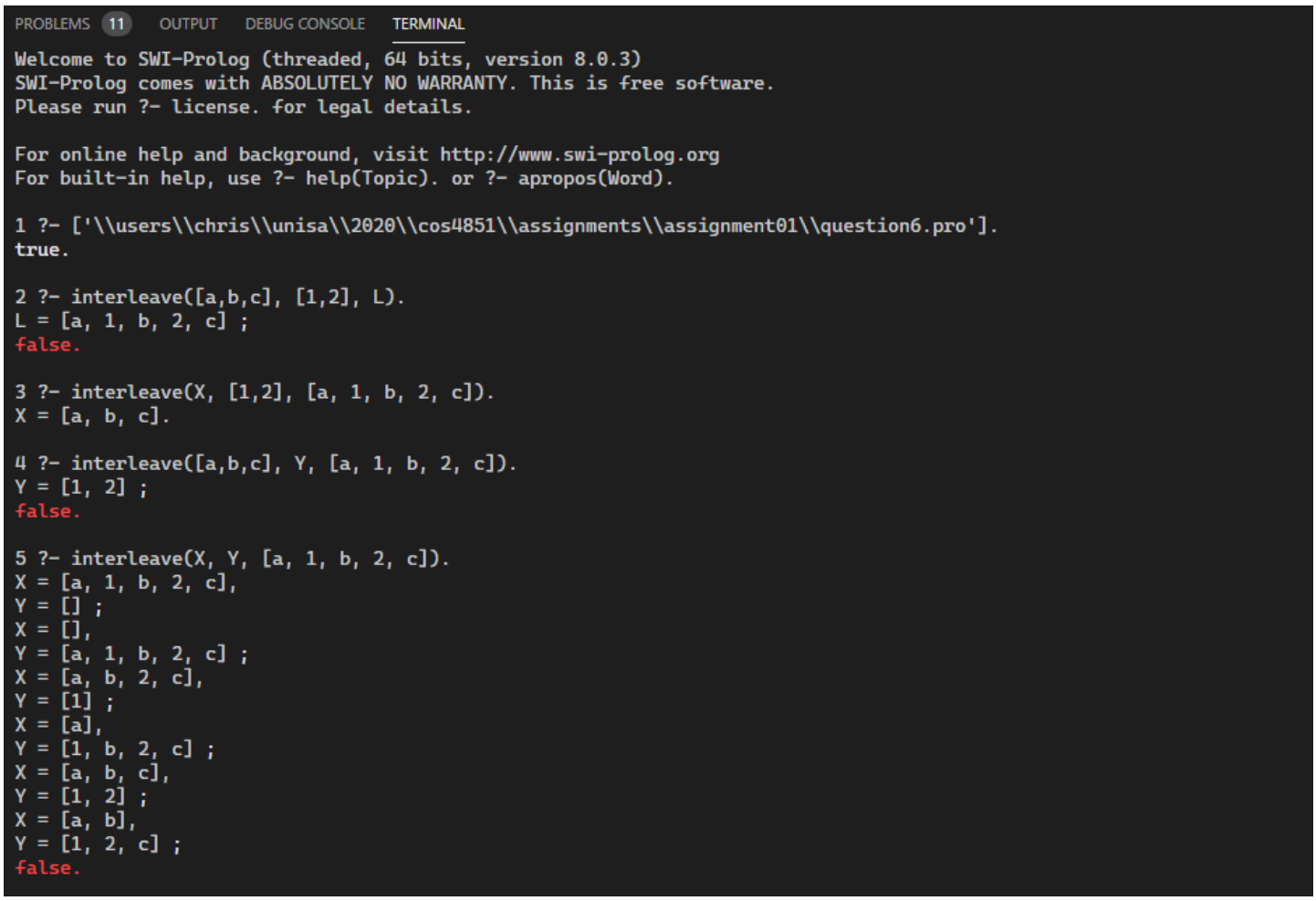
Yes, `[a, pred_b(1, mill_14), b, pred_c(cdc,[8, 9]), d, e]` and `[a, X, b, pred_c(Y,Z), d, e]` match.
`X` can be instantiated to `pred_b(1, mill_14)`, `Y` can be instantiated to `cdc` and `Z` can be instantiated to `[8, 9]`.

Question 6

6 (a)

```
interleave([],[],[]).

interleave([H|T],[],[H|T]).

interleave([],[H|T],[H|T]).

interleave([L1H|L1T], [L2H|L2T], [L1H,L2H|L]) :-
    interleave(L1T, L2T, L).
```

Yes, the input/output roles are reversible, as can be seen by execution 3, 4 and 5 in the screenshot below:



6 (b)

```
transpose([],[],[]).

transpose([L1H|L1T], [L2H|L2T], [(L1H,L2H)|L]) :-
    transpose(L1T, L2T, L).
```

Yes, the input/output roles are reversible, but the transpose function requires that the two input lists be of the same size.

```
PROBLEMS 13  OUTPUT  DEBUG CONSOLE  TERMINAL

Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- ['c:\\Users\\chris\\unisa\\2020\\COS4851\\Assignments\\assignment01\\question6.pro'].
true.

2 ?- transpose([a,b,c,d], [1,2,3,4], L).
L = [(a, 1),  (b, 2),  (c, 3),  (d, 4)].

3 ?- transpose(X, [1,2,3,4], [(a, 1),  (b, 2),  (c, 3),  (d, 4)]).
X = [a, b, c, d].

4 ?- transpose([a,b,c,d], Y, [(a, 1),  (b, 2),  (c, 3),  (d, 4)]).
Y = [1, 2, 3, 4].

5 ?- transpose(X, Y, [(a, 1),  (b, 2),  (c, 3),  (d, 4)]).
X = [a, b, c, d],
Y = [1, 2, 3, 4].

6 ?-
```

6 (c)

```
inner_prod([],[],0).

inner_prod([L1H|L1T], [L2H|L2T], Result) :-
    inner_prod(L1T, L2T, RecursiveResult),
    Result is L1H * L2H + RecursiveResult.
```

No, the input/output roles are not reversible. The this function reduces two vectors of the same length down to a single value which has no deterministic way of being reversed.

```
PROBLEMS 13  OUTPUT  DEBUG CONSOLE  TERMINAL

Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- ['c:\\Users\\chris\\unisa\\2020\\COS4851\\Assignments\\assignment01\\question6.pro'].
true.

2 ?- inner_prod([2,3,3,2],[1,2,3,4], R).
R = 25.

3 ?- inner_prod([1,1,1,1],[1,2,3,4], R).
R = 10.

4 ?- inner_prod([4,3,2,1],[1,2,3,4], R).
R = 20.

5 ?-
```