README.md

Prolog Assignment

Instructions

Do all 5 for an A:

1. structures parent(X,Y), male(X), and female(X), write a structure that defines mother(X,Y).
2. Using the structures parent(X,Y), male(X), and female(X), write a structure that defines sister(X,Y).
3. Write a predicate second(X,List) that checks wether X is the second element of List.
4. Write a predicate twice(n,Out) whose left argument is a list, and whose right argument is a list consisting of every element in the list written twice.

```
twice([a,4,buggle], X).
X=([a,a,4,4,buggle,buggle]).

twice(W,[b,b,a,a]).
W=[b,a].
```

5. Write 3-parameter predicate interleave which takes three lists as arguments and combines the elements of the first two into the third as follows:

```
?-interleave([a,b,c],[d,e,f],X).
X=[a,d,b,e,c,f]
```

Results

For each, hand in a print-out showing your definition and results with enough test cases to demonstrate the correct behavior.

Male, Female and Parent Structures:

```
male(mason).  %parent
male(ealair).
male(madhav).

female(ayame). %parent
female(regina).
female(vera).

parent(mason,ealair).
parent(mason,madhav).
parent(mason,regina).
parent(mason,vera).

parent(ayame,ealair).
parent(ayame,madhav).
parent(ayame,regina).
parent(ayame,vera).
```

1. Mother Structure

```
mother(X,Y) :- parent(X,Y),female(X).
```

```
For online help and background, visit http://www.swi-prolog.org
X = mason ;
X = ealair ;
X = madhav.

?- female(Y).
Y = ayame ;
Y = regina ;
Y = vera.

?- mother(ayame,X).
X = ealair ;
X = madhav ;
X = regina ;
X = vera.

?- mother(ayame,ealair).
true .

?- mother(ayame,vera).
true.

?- mother(ayame,mason).
false.
```

2. Sibling and Sister Structure

```
sibling(X,Y) :- parent(W,X),parent(W,Y),not(X=Y).
sister(X,Y) :- sibling(X,Y),female(X).
```

```
?- male(X).
X = mason ;
X = ealair ;
X = madhav.

?- female(Y).
Y = ayame ;
Y = regina ;
Y = vera.

?- sister(vera,mason).
false.

?- sister(vera,regina).
true .

?- sister(vera,ayame).
false.
```

3. Second Element

```
second(X,[_|[X|_]]).
```

```
?- second(1,[1,2,3,4,5]).
false.

?- second(2,[1,2,3,4,5]).
true.

?- second(3,[1,2,3,4,5]).
false.

?- second(a,[1,a,3,4,5]).
true.
```

4. Twice

```
twice([],[]).
twice([X|Y],[X,X|Z]) :- twice(X,Z).
```

```
?- twice([a,b],X).
X = [a, a, b, b].

?- twice([a,4,buggle],X).
X = [a, a, 4, 4, buggle, buggle].

?- twice(W,[b,b,a,a]).
W = [b, a].

?- twice(W,[b,b,b,b,a,a,a,a]).
W = [b, b, a, a].
```

5. Interleave

```
interleave([],[],[]).
interleave([],[H2|T2],[H2|T2]).
interleave([H1|T1],[],[H1|T1]).
interleave([H1|T1],[H2|T2],[H1,H2|TT]) :- interleave(T1,T2,TT).
```

```
?- interleave([a],[b],X).
X = [a, b] .

?- interleave([a,c],[b,d],X).
X = [a, b, c, d] .

?- interleave([a,b,c],[d,e,f],X).
X = [a, d, b, e, c, f] .
```