Assignment - 08 [Introduction to PROLOG]

a)

```
female (pam).
female (liz).
female (pat).
female (ann).
male (jim).
male (bob).
male (tom).
male (peter).
male (at).
parent (pam, bob).
parent (tom, bob).
parent (tom, liz).
parent (bob, ann).
parent (bob, pat).
parent (pat, jim).
parent (bob, peter).
parent (peter, jim).
mother (X, Y) :- parent (X, Y)., female (X).
father (X, Y) :- parent (X, Y), male (X).
haschild (X) :- parent (X, _).
sister (X, Y) :- parent (Z, X), parent (Z, Y), female (X), X\==Y.
brother (X, Y) :- parent (Z, X), parent (Z, Y), male (X), X\==Y.
```

2)

```
obs(tree).
obs(human).
obs(car).
obs(roadblock).
brakes(X) :- obs(X).
acc(X) :- !, obs(X).
```

3)

```
father(a, b).
father(a, c).
father(b, d).
father(b, e).
father(c, f).
brother(X, Y) :- father(Z, X), father(Z, Y), not(X = Y).
cousin(X, Y) :- father(Z, X), father(W, Y), brother(Z, W).
grandson(X, Y) :- father(Z, X), father(Y, Z).
descendent(X, Y) :- father(Y, X).
descendent(X, Y) :- father(Z, X), descendent(Z, Y).
```

4)

```
% Define a predicate quicksort(L,K) which, given a list of integers L,
% returns an ordered list K obtained from L with the method of
% quicksort. Solution
quickSort([],[]).
quickSort([X],[L],[K]) :- split(X, L, L1, L2),
                          quicksort(L1, K1),
                          quicksort(L2, K2),
                          append(K1, [X|K2], K).
split(_,[],[],[]).
split(X,[Y|L],K,[Y|M]) :- X < Y, split(X, L, K, M).
split(X,[Y|L],[Y|K],M) :- X >= Y, split(X, L, K, M).
```

1) Output

brother (jim, at)

-> true

father (onn, male)

-> false.

2) Output

occ(car).

-> true.

occ(tree)

-> false.

3) Output

grandson(d, a).

->true

Cousin(d, c).

-> false.

4) quicksort([2, 5, 1, 8, 36], [1, 2, 3, 5, 6, 8])

-> true.

-> false

quicksort([2, 5, 1, 8, 3, 6], [2, 1, 3, 5, 6, 8])

-> false.