

Practical No 10**Date:** 17/08 /2022**Title:** Write a Prolog program that captures basic family relations.**Description:**

Prolog - PROgramming in LOGic, was first developed by Alain Colmerauer and Philippe Roussel in 1972 - A Logic Programming language based on the predicate logic.

In prolog, clauses are actually descriptive statements that specifies what is true about the problem and because of that Prolog is also known as declarative language or rule-based language.

Basically, in prolog program we write the program statements in terms of facts and rules. The system reads in the program and stores it. Upon asking the questions (known as queries) the system gives the answer by searching through the possible solution(s).

- Facts-Something that is declared to always be true.

Ex-father(bob,george)

mother(alice,jeffrey)

- Rules- Something which is true only if all conditions (sub goals) on the right are true

Ex-

parent(X,Y) :- father(X,Y).

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

grandparent(X,Z) :- parent(X,Y), parent(Y,Z).

Program Code:

```

1 parent(carl,beth).
2 parent(carl,derryl).
3 parent(amy,beth).
4 parent(amy,derryl).
5 parent(derryl,jon).
6 parent(beth,jane).
7 parent(dan,carl).
8 parent(peter,amy).
9 parent(peter,jan).
10 parent(jan,lucy).
11 parent(stacy,amy).
12 parent(stacy,jan).
13
14
15 father(X,Y):-parent(X,Y),male(X).
16 mother(X,Y):-parent(X,Y),female(X).
17 grandparent(X,Z):-parent(X,Y),parent(Y,Z).
18 sibling(X,Y):-parent(Z,X),parent(Z,Y), X \= Y.
19 cousin(X,Y):- grandparent(Z,X),grandparent(Z,Y), X \
20 grandpa(X,Z):-grandparent(X,Z), male(X).
21 grandma(X,Z):-grandparent(X,Z),female(X).
22 sister(X,Y):-sibling(X,Y),female(X).
23 brother(X,Y):-sibling(X,Y),male(X).
24 uncle(Z,Y):-sibling(Z,X),male(Z),mother(X,Y).
25 aunt(Z,Y):-sibling(Z,X),female(Z),mother(X,Y).
26 niece(X,Y):-uncle(Y,X);aunt(Y,X),female(X).
27 nephew(X,Y):-uncle(Y,X);aunt(Y,X),male(X).
28
29
30
31


```

```


male(carl).
male(derryl).
male(jon).
male(peter).
male(dan).
female(amy).
female(beth).
female(jane).
female(jan).
female(lucy).
female(stacy).

```


Input and Output

 *father*(X,jon).

X = derryl

 *cousin*(X,jane).


X = jon

 *mother*(amy,Y).


Y = beth

 *grandpa*(X,lucy).


X = peter

 *sister*(X,Y).

X = beth,
Y = derryl

 *brother*(X,Y).

X = derryl,
Y = beth

 *uncle*(X,jane).

X = derryl

 *aunt*(X,Y).

X = amy,
Y = lucy
X = jan,
Y = beth
X = jan,
Y = derryl

Next 10 100 1,000 Stop

 *niece*(X,jan).

X = beth

Next 10 100 1,000 Stop

Conclusion: Thus we have created the knowledge base to capture basic family relations in Prolog.

Practice programs:

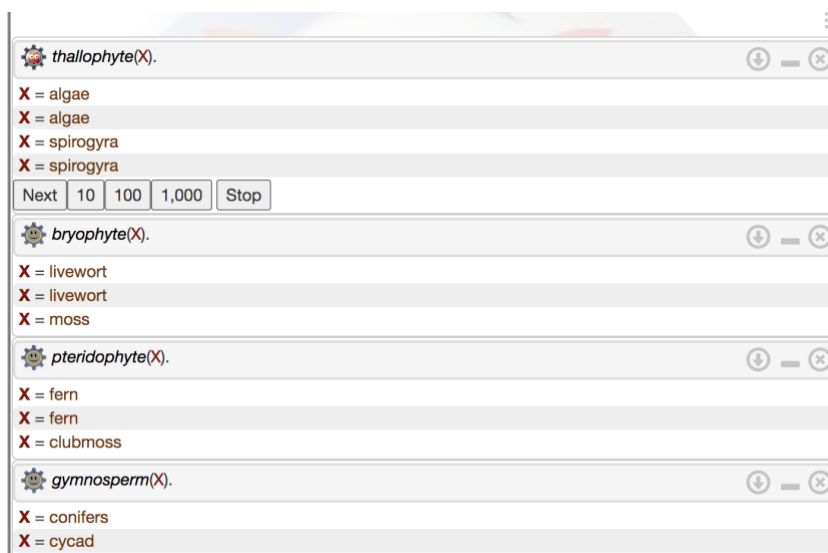
1. Create a knowledge base in Prolog which Captures the plant kingdom.
2. Create a knowledge base in Prolog which suggests the jobs based on qualification and experience.

1) PLANT KINGDOM

```

1 bodytype(algae,primitive).
2 bodytype(algae,unbranched).
3 bodytype(spirogyra,filamental).
4 bodytype(spirogyra,unbranched).
5 bodytype(fern,stem).
6 bodytype(fern,leaves).
7 bodytype(conifers,vascular).
8 bodytype(conifers,differentiated).
9 bodytype(cycad,vascular).
10 bodytype(cycad,differentiated).
11 characteristics(livewort,nonvascular).
12 characteristics(livewort,seedless).
13 characteristics(moss,perennial).
14 characteristics(moss,autotrophic).
15 location(clubmoss,shady).
16 location(clubmoss,damp).
17 location(cycad,forests).
18
19
20 thallophyte(X):-bodytype(X,primitive);bodytype(X,unbranched);bodytype(X,filamental);bodytype(X,unbranched).
21 bryophyte(X):-characteristics(X,nonvascular);characteristics(X,seedless);characteristics(X,perennial);characteristics(X,autotrophic).
22 pteridophyte(X):-bodytype(X,stem);bodytype(X,leaves);location(X,shady);location(X,damp).
23 gymnosperm(X):-bodytype(X,vascular);bodytype(X,differentiated);location(X,forests).
24
25
26

```



2) JOB SUGGESTIONS

```

1 job(manager, bachelor, experienced).
2 job(engineer, bachelor, fresher).
3 job(teacher, bachelor, experienced).
4 job(programmer, bachelor, moderate).
5 job(manager, master, moderate).
6 job(doctor, master, experienced).
7 job(researcher, master, experienced).
8 job(accountant, bachelor, moderate).
9 job(architect, master, fresher).
10
11 suggest(X, Job) :-qualification(X, Qualification),experience(X, Experience),job(Job, Qualification, Experience).
12
13 qualification(j, bachelor).
14 qualification(k, master).
15 qualification(l, bachelor).
16 qualification(m, bachelor).
17 qualification(n, master).
18 qualification(o, master).
19 experience(j, experienced).
20 experience(k, fresher).
21 experience(l, moderate).
22 experience(m, fresher).
23 experience(n, experienced).
24 experience(o, moderate).
25
26
27
28
29
30
31

```

suggest(j,Job).

Job = manager

Job = teacher

suggest(k,Job).

Job = architect

suggest(l,Job).

Job = programmer

Job = accountant

suggest(m,Job).

Job = engineer

suggest(n,Job).

Job = doctor

suggest(o,Job).

Job = manager