

# Prolog Programming Assignment

NAME:-

Khan Zarrar Hussain Lakir Hussain

Roll No.r (29)

Class:- BE - IT

Batch : I<sub>2</sub>

Subject:- Artificial Intelligence (ISLAB)

DOP

DOA

Marks

Sign.

① How does the queries in kb.pl file are executed?

→ code:  
loves(vincent, mia)  
loves(marcellus, mia)  
loves(pumpkin, honey-bunny)  
loves(honey-bunny, pumpkin)

jealous(x, y) :-  
loves(x, z)  
loves(y, z)

Query 1: ? - loves(x, mia)

Output:

x = vincent  
x = marcellus

Explanation: Here as we know vincent loves mia as well as marcellus loves mia. Thus the kb assumes that x is either vincent or marcellus.

Query 2: ? - jealous(x, y)

Output: x = y, x = vincent  
x = vincent

x = Marcellus

x = y, y = Marcellus

x = y, y = Pumpkin

x = y, y = Honey-bunny

Explanation: As there is no fixed parameter

in our query. The query will produce output of every jealous ( $x, y$ ) pair on our prolog code. The jealous() rule follows jealous( $x, y$ ) :- loves( $x, z$ ), loves( $y, z$ ).

Initially,  $x$  &  $y$  both were associated to vincent i.e. self association. It then follows reflexive property for the rest of the prolog code.

② How does the queries in lists.pl file are executed?

→ Code : suffix( $X_s, Y_s$ ) :-  
append(-,  $Y_s, X_s$ )

~~suffix~~  $\rightarrow$  sublist( $X_s, Y_s$ ) :-

~~suffix~~  $\rightarrow$  suffix( $X_s, Z_s$ ),  
~~suffix~~  $\rightarrow$  prefix( $Z_s, Y_s$ )

Prefix( $X_s, Y_s$ ) :-

append( $Y_s, - , X_s$ )

nrev([ ], [ ])

nrev([ H | T0 ], L) :-

nrev(T0, T),

append(T, [H], L)

Query 1 : ? - sublist([a, b, c, d, e], [c, d]).

Output : True

Explanation : A sublist procedure looks for a match between the first elements of the sublists & the main-list.

Here,  $[c, d]$  is the sub-list of the main list  $[a, b, c, d, e]$ . As the main list contains the sublist  $[c, d]$ , the output is true. Else, the output would have been false.

Query 2: ? - suffix([a,b,c], Zs)

Output:  $Zs = [a, b, c]$

$Zs = [b, c]$

$Zs = [c]$

$Zs = []$

False

Explanation: Suffix in general eliminates the front elements from a list.

Here, by using suffix procedure,  $[a, b, c]$  elements are removed from a and continuous until all the elements are removed. As there are no more elements in the list, the output will be displayed as 'false'.

(3) Programming: create a prolog code to find factorial of a number

Code: factorial(0, 1).  
factorial(N, F) :-

$N > 0$

$N$ , is  $N - 1$ ,

factorial( $N_1$ ,  $F_1$ ),

$N$ , is  $N * F_1$ ,

Page No.	
Date	

Query :- factorial(3, w).

Output: w=6

(4) In example dataset movies.pl write query strings and results of query execution for any of 5 tasks:

a) In which year was the movie American Beauty released?

Query: ?- movie(American-Beauty, y)

Output: y = 1999

b) find the movies released in year 2000.

Query: ?- movie(M, 2000)

Output: M = down-from-the-mountain  
M = o-brother-where-art-thou  
M = ghost-world

c) find movies released before 2000.

Query: ?- movie(M, y), y < 2000

Output: M = american-beauty  
y = 1999

M = anna

y = 1987

M = borton-fink

y = 1991

d) find the movies released after 1990. (A)

Query: ? - movie(M, y), y > 1990

Output: M = american-beauty  
y = 1999

M = barton-fink  
y = 1991

e) Find a director of a movie in which  
Scarlett Johansson appeared. (B)

Query: ? - actress(M, scarlett.johansson),  
director(M, D)

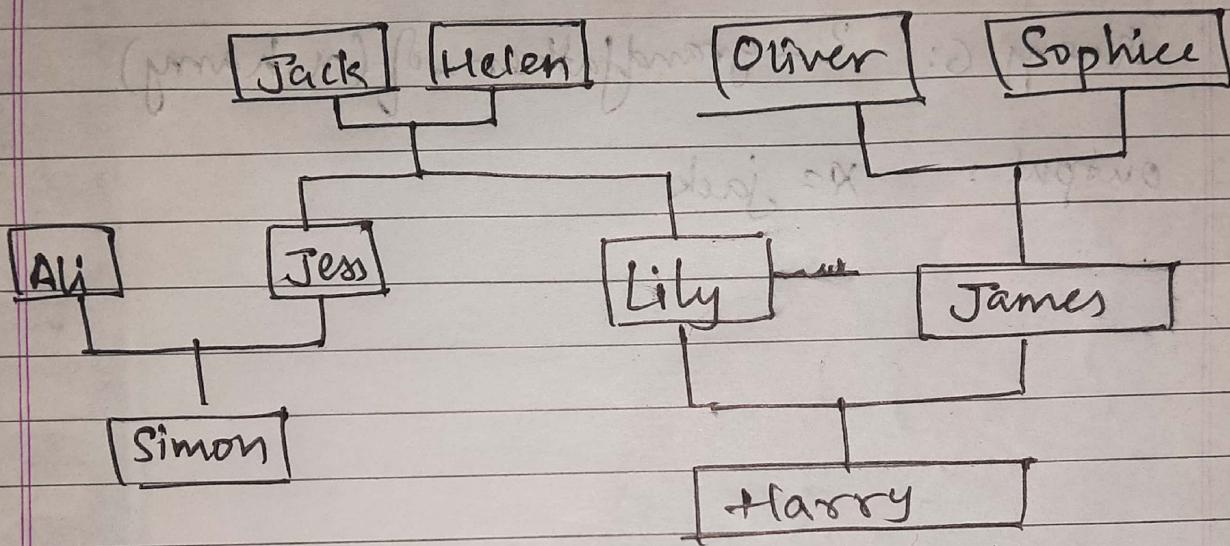
D = peter-webber  
M = girl-with-a-pearl-earring

(5)

Draw a family tree of any arbitrary family which has the following relations: mother, father, daughter, son, grandmother, sibling, uncle, person, male, female. You need to convert it into KB & write 6 queries results on your KB.



Diagram:



Family Tree

Query 1: ? - mother-of ( $x$ , jess)

Output:  $x = \text{helen}$

Query 2: ? - parent-of ( $x$ , simon)

Output:  $x = \text{jess}$

Query 3: ? - sister-of ( $x$ , lily)

Output:  $x = \text{jess}$

Query 4: ? = parent-of(x, harry).

Output: x = lily

x = james

Query 5: ? = aunt-of(x, simon)

Output: x = lily

Query 6: ? = grandfather-of(x, harry)

Output: x = jack

lance

[lily]

[jack]

[simon]

lance

[simon]

lance

(wife(x)) for lance -? : lily

wife(x) : lily

(son(x)) for lance -? : jack

son(x) : jack

(wife(x)) for jack -? : lily

wife(x) : lily