

EX.NO: 07

INTRODUCTION TO PROLOG

AIM :

To learn PROLOG terminologies and write basic programs.

TERMINOLOGIES:

1. Atomic Terms: -

Atomic terms are usually strings made up of lower- and uppercase letters, digits, and the underscore, starting with a lowercase letter.

Ex:

dog
ab_c_321

2. Variables: -

Variables are strings of letters, digits, and the underscore, starting with a capital letter or an underscore.

Ex:

Dog
Apple_420

3. Compound Terms: -

Compound terms are made up of a PROLOG atom and a number of arguments (PROLOG terms, i.e., atoms, numbers, variables, or other compound terms) enclosed in parentheses and separated by commas.

Ex:

is_bigger(elephant,X)
f(g(X,_),7)

4. Facts: -

A fact is a predicate followed by a dot.

Ex:

bigger_animal(whale).
life_is_beautiful.

5. Rules: -

A rule consists of a head (a predicate) and a body (a sequence of predicates separated by comma)

CODE :

**SWISH** File▼ Edit▼ Examples▼ Help▼
Program x +
1 likes(john,brittney).
2 likes(dan,sally).
3 likes(sally,dan).
4 married(X,Y) :- likes(X,Y) , likes(Y,X).
5 friends(X,Y) :- likes(X,Y) ; likes(Y,X).
6

OUTPUT :

GIVING QUERY:

?- friends(X,Y).

Examples▲ History▲ Solutions▲

☐ table results **Run!**

The screenshot displays a Prolog interpreter interface with four stacked query windows. Each window has a title bar with a gear icon, a query, and control buttons (Next, 10, 100, 1,000, Stop). The queries and their results are as follows:

- Query 1:** `likes(john,Y).`
Result: `Y = brittney`
- Query 2:** `likes(Y,brittney).`
Result: `Y = john`
- Query 3:** `married(X,Y).`
Result: `X = dan,`
`Y = sally`
- Query 4:** `friends(X,Y).`
Result: `X = john,`
`Y = brittney`

RESULT :

Thus, the implementing of PROLOG terminologies is successfully executed and the output is verified.