

Name : TARRUN.B

Reg no : 2116241801291

Exp no : 5

Exp name : decision making and knowledge representation

Date : 10.05.2025

Program :

% Given facts

likes(mary, food).

likes(mary, wine).

likes(john, wine).

likes(john, mary).

% Rules based on the conditions:

likes(john, X) :- likes(mary, X). % John likes anything that Mary likes

likes(john, Y) :- likes(Y, wine). % John likes anyone who likes wine

likes(john, Y) :- likes(Y, Y). % John likes anyone who likes themselves

% Sample queries:

% Query 1: Does John like food?

% ?- likes(john, food).

% Query 2: Does John like wine?

% ?- likes(john, wine).

% Query 3: Does John like food if Mary likes food?

% ?- likes(john, food).

% Query 4: Who does John like?

% ?- likes(john, Y).

Output :

The screenshot shows the SWISH Prolog IDE interface. On the left, a program is loaded with the following rules:

```
1 likes(mary, food).
2 likes(mary, wine).
3 likes(john, wine).
4 likes(john, mary).
5 likes(john, X) :- likes(mary, X).
6 likes(john, Y) :- likes(Y, wine).
7 likes(john, Y) :- likes(Y, Y).
```

On the right, the output window displays the results of two queries. The first query is `likes(john, food).`, which returns `true`. The second query is `likes(john, Y).`, which returns a table of results for `Y`:

Y	
wine	1
mary	2
food	3

Below the table, there are buttons for `Next`, `10`, `100`, `1,000`, and `Stop`. At the bottom of the output window, there are tabs for `Examples`, `History`, and `Solutions`, along with a checkbox for `table results` and a `Run!` button.