Ex No: 2

IMPLEMENTATION OF DECISION MAKING AND KNOWLEDGE REPRESENTATION

Aim:

To implement decision making and knowledge representation using prolog tool.

Prolog Code:

```
% Rule to find the minimum of two numbers
minimum(X, Y, X) :- X =< Y. % If X is less than or equal to Y, X is the minimum.
minimum(X, Y, Y) :- X > Y. % If X is greater than Y, Y is the minimum.

% Rule to find the maximum of two numbers maximum(X, Y, X) :- X >= Y. % If X is
greater than or equal to Y, X is the maximum. maximum(X, Y, Y) :- X < Y. % If X is less
than Y, Y is the maximum.
```

Example Queries:

1. To find the minimum of two numbers:

```
?- minimum(5, 10, Min).
```

Output:

```
Min = 5.
```

2. To find the maximum of two numbers:

```
?- maximum(5, 10, Max).
```

Output:

```
Max = 10.
```

```
?- minimum(8, 3, Min), maximum(8, 3, Max).
```

Output:

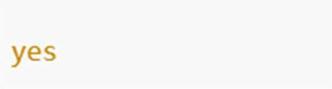
```
Min = 3, Max = 8.
```

Prolog Code:

```
% 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).
```

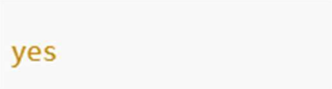
Output:

Query: `?- likes(john, food).`



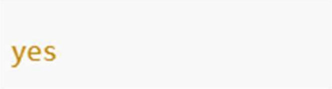
```
yes
```

Query: `?- likes(john, wine).`



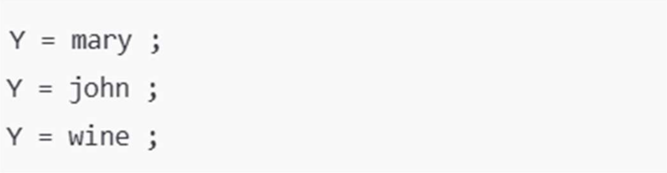
```
yes
```

Query: `?- likes(john, food).`



```
yes
```

Query: `?- likes(john, Y).`



```
Y = mary ;
Y = john ;
Y = wine ;
```

Query?- likes(john, Y).

Y = mary ;

Y = john ;

Y = wine ;