Program 29
Forward chaining.

Code:-

% Define facts representing relationships between animals and their characteristics

```
mammal(dog).

mammal(cat).

bird(parrot).

bird(eagle).

has_feathers(parrot).

has_feathers(eagle).

has_fur(dog).

has_fur(cat).


% Define rules for inferring additional information

can_fly(X) :- bird(X), has_feathers(X).

has_fur_animal(X) :- mammal(X), has_fur(X).


% Forward chaining algorithm

forward_chaining :-
   repeat,
   (  new_fact(X),        % Look for new facts

      asserta(X),         % Add new facts to the database
```

```prolog
        fail            % Fail to loop again and find more facts
    ;   !,              % Cut to prevent backtracking
        write('No more new facts to infer.'), nl
    ).


% Define rules for inferring new facts
new_fact(can_fly(X)) :-
    bird(X),
    \+ can_fly(X).


new_fact(has_fur_animal(X)) :-
    mammal(X),
    has_fur(X),
    \+ has_fur_animal(X).


% Example queries
% ?- forward_chaining.
```
OUTPUT:

```
GNU Prolog console

File   Edit   Terminal   Prolog   Help

GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('D:/GNU-Prolog/bin/forward.pl').
compiling D:/GNU-Prolog/bin/forward.pl for byte code...
D:/GNU-Prolog/bin/forward.pl compiled, 36 lines read - 3021 bytes written, 11 ms

yes
| ?- forward_chaining.
No more new facts to infer.

yes
| ?-
```

% Example queries

% ?- forward_chaining.

OUTPUT: