Write a Prolog Program for forward Chaining. Incorporate required queries.

AIM

To write a Prolog program that uses forward chaining to infer new facts from existing facts and rules, such as determining grandparents or ancestors from parent relationships.

ALGORITHM

1. Start the program.
2. Define base facts using parent(X, Y) to represent direct relationships.
3. Define inference rules like grandparent(X, Z) and ancestor(X, Z) using existing facts.
4. Load the program into the Prolog interpreter.
5. Query the program using statements like grandparent(X, Z) or ancestor(X, Z) to infer new relationships.
6. Prolog automatically applies forward chaining to derive all possible facts from the rules and given facts.
7. Stop.

```
% Facts
parent(john, mary).
parent(mary, susan).
parent(susan, tom).

% Rules (forward chaining: infer grandparent from parent facts)
grandparent(X, Z) :- parent(X, Y), parent(Y, Z).

% Additional rules
ancestor(X, Z) :- parent(X, Z).
ancestor(X, Z) :- parent(X, Y), ancestor(Y, Z).
```

OUTPUT:

```
?-
% c:/Users/gayathri/Downloads/parent.pl compiled 0.00 sec, 6 clauses
?- grandparent(X, Z).
X = john,
Z = susan ;

?- ancestor(X, tom).
X = susan ;
X = john ;
X = mary ;
false.

?- ancestor(mary, tom).
true
randparent(john, tom).Unknown action: g (h for help)
Action? ;

?- grandparent(john, tom).
false.

?-
```

RESULT

The program successfully infers new relationships using forward chaining. For example, it identifies john as the grandparent of tom and lists all ancestors of tom based on the parent relationships.