| Experiment No.7 |
| Study and create knowledge base in Prolog. |
| Date of Performance: |
| Date of Submission: |

Aim: Study and create knowledge base in Prolog.

Objective: To study and use AI programming language to create knowledge base.

Theory: Take any problem and represent the knowledge (facts) in prolog. Also you can use this for reasoning purpose.

Example: The problem of murder mystery.

Five persons Alice, her husband, brother, son and daughter

Event: One murder. One of the five is victim and one is Killer.

Rules:

1) Husband and Alice was not together on the night of murder.

2) The killer and victim were on the beach.

3) On the night of murder, one male and one female was in the bar.

4) The victim was twin and the counterpart was innocent.

5) The killer was younger than the victim.

6) One child was alone at home.

Code for Prolog problem of murder mystery in Artificial Intelligence

```
predicates
%   pair(symbol,symbol)
     iskiller(symbol,symbol)
    male(symbol)
    female(symbol)
    isvictim(symbol)
    not_at_bar(symbol,symbol)
    not_at_beach(symbol,symbol)
    not_alone(symbol)
    twin(symbol,symbol)
    younger(symbol,symbol)
    child(symbol)

clauses
    male(husband).
    male(brother).
    male(son).

    female(alice).
    female(daughter).
```

twin(brother,alice).
twin(son,daughter).

child(son).
child(daughter).

EXAMPLE:

Problem: Family Relationships

Facts:
1. John is the father of Mary.
2. Mary is the mother of Tim.
3. Tim is the brother of Sarah.
4. Sarah is the sister of Mary.
5. Jack is married to Mary.

Rules:
1. If X is the father of Y, then X is a male.
2. If X is the mother of Y, then X is a female.
3. If X is married to Y, then Y is married to X (assuming marriage is symmetrical).
4. If X is the parent of Y, then X is older than Y.
5. Siblings share at least one parent.

Prolog Representation:

```prolog
% Facts
male(john).
female(mary).
male(tim).
female(sarah).
male(jack).

father(john, mary).
mother(mary, tim).
mother(mary, sarah).
married(jack, mary).

% Rules
parent(X, Y) :- father(X, Y); mother(X, Y).
older(X, Y) :- parent(X, Y).
```

```
sibling(X, Y) :-
    parent(Z, X),
    parent(Z, Y),
    X \= Y.
```

Conclusion: In conclusion, Prolog provides a powerful framework for representing and reasoning about complex problems through the use of facts and rules. By encoding knowledge in a declarative manner, Prolog allows for efficient querying and logical inference, making it suitable for various domains such as artificial intelligence, natural language processing, and expert systems. Through the examples provided, we've demonstrated how Prolog can be employed to model intricate scenarios, facilitating the deduction of conclusions based on specified rules and relationships. Whether it's solving murder mysteries or analyzing family dynamics, Prolog's ability to represent knowledge and perform reasoning makes it a valuable tool in the realm of computational logic and problem-solving.