Experiment -1

Problem Statement:

practice on Facts and Queries

Program:

```
> likes (ram, seeta).
  likes (sajeed, razi).
```

output:

```
? likes (ram, seeta).
  yes
? likes (sajeed, razi).
  yes
> likes (ram, mango).
  girl (seema).
% likes (bill, cindy).
  toad (rose).
  owns (john, gold).
```

output:

```
? likes (ram, mango).
  yes
? toad (rose).
  yes
? owns (john, gold).
  yes
? girl (seema).
  yes
```

? likes(ram, mango).
no

> friends(Krushna, rav).
boy(archit).
teachers(Krushna, Prolog).

output:

? teachers(Krushna, prolog).
Yes.

? boy(archit).

Yes

Description:

Here we convert the basic facts into Queries by using prolog.

Ex: Ram likes Seeta → (Fact)

likes(Ram, seeta). → (Query)

Experiment 2:

Problem Statement:

performing arithmetic Operations using prolog.

program:

```
> Sum(A,B):-
    S is A+B,
    write(S).

-? Sum(10,15).
    25
    Yes
```

```
> Sub(A,B):-
    S is A-B,
    write(S).

-? Sub(15,10).
    5
    Yes
```

```
> multi(A,B):-
    M is A*B,
    write(M).

-? multi(10,3)
    30
    Yes
```

```
> pow(A,B):-
    P is A**B,
    write(P).

-? pow(2,3).
    8
    Yes
```

```
> div(A,B):-
    V is A/B,
    write(V).

-? div(10,2).
    5
    yes
```

```
> idiv(A,B):-
    IV is A//B,
    write(IV).

-? idiv(10,2).
    0
    Yes
```

Description:

Arithmetic Operations can be performed using prolog functions by giving numericals while execution operations like addition, subtraction, multiplication, division, modular division, power can be performed.

Experiment3:

Problem Statement:

To calculate the Intrest & discount with principle Amount, tenure & rate given.

Program

```
Si(P,T,R):-
    I is (P*T*R)/100,
    AfterDiscount is (2*I)/100,
    write(I),nl,
    write(AfterDiscount).
```

Output:

```
Si(1000,2,2).
40.0
0.800000004
Yes
```

Description:

This program allows us to calculate the Simple Intrest of the given Amount and the Final Amount obtained after discount using prolog.

Experiment 4:

Problem Statement:

Binov's Home is located exactly in the middle between Ajay and Vijay's Ordinators. Find out where Binov's house is located.

Program:

```
loc(x1,y1,x2,y2):-
    x is (x1+x2)/2,
    y is (y1+y2)/2,
    write('Binov's house is located at:'),
    write('('),
    write(x),
    write(','),
    write(y),
    write(')').
```

Output:

```
loc(2,5,3,4)
Binov's house is located at (2.5,4.5)
yes
```

Description:

Program tells us about how to calculate the coordinates of the centered points when two points / coordinates are given.

Experiment 5:

Problem Statement:

Conversion of temperature in centigrades into fahrenheit and checking whether it is below freezing point.

Program:

```
tempconv(C):-
  F is 9/5*(C+32),
  write('the temperature'),
  write(C),
  Write ('centigrade'),
  Write(F),
  Write('Farenheit'),
  F<0,
  Write('the temp is below freezing').
```

Output:

```
tempconv(35)
35 centigrade is 120.60000 farenheit
the temperature is below freezing point.
Yes.
```

Description:

Here we convert the temperature from centigrade to farenheit and also checks whether it is below freezing or not.

Experiment 6:

Problem Statement:

To Calculate values using relational Operations.

Program:

```
greaterthan(A,B):-
    Write('First is greater than Second'),nl,
    A>B
```

O/P: greaterthan(5,3).

First is greater than Second

Yes

```
lessthan(A,B):-
    Write('First is less than Second"),nl,
    A<B
```

O/P: lessthan(1,6)

First is less than Second

Yes

```
equalto(A,B):-
    Write('First & Second are equal'),nl,
    A==B
```

O/P: equalto(5,5)

First & Second are equal

```
notequalto(A,B):-
  write("First and second are not same"),nl,
   A!=B
```

o/p:

```
notequalto(5,7)
  First and second are not same
  yes
```

Description:

Relational operators in prolog compares values (or) expressions, determining the relationship like '>, <, >=, <=, ==, !='. They are essential for establishing logical conditions and constraints in prolog programs.

Experiment 7:

Problem Statement:

To Find the Factorial of a given number.

Program:

```
fact(N,F):-
  write('the factorial for'),
  write(N),
  write('is'),
  fact(N,F1),
    F is F1,
    write(F).

fact(0,1).
fact(1,1).

fact(N,F):-
  N>1
  N1 is N-1
  fact(N1,F1)
   F is N*F1
```

Output:

```
fact(4,F)
the Factorial for 4 is 24
F=24?
yes
```

Description: Factorial of a number in prolog is computed recursively by multiplying the number with the factorial of its predecessor until reaching 1, leveraging predicates and pattern matching for implementation.

Experiment 8:

Problem Statement:

Write a program in prolog to Solve Tower of Hanoi.

Program:

```
move(1,X,Y,_):-
    write('move top disk from'),
    write(X),
    write('to'),
    write(Y),
    nl
    move(N,X,Y,Z):-
    N>1
    M is N-1,
    move(M,X,Z,Y),
    move(1,X,Y,_),
    move(M,Z,Y,X).
```

Output:

```
move(3, Source, target, Auxillary).
    move top disk from Source to target
                       Source to Auxillary.
                       target to Auxillary
                       Source to target
                       Auxilary to Source
                       Auxillary to target
                         Source to target.
```

Description: In prolog, the Towers of Hanoi problem is elegantly solved through recursive predicates, where each move abides the constraints of the puzzle, ensuring no larger disk is placed at top of smaller one while transfering the entire stack from one rod to another