PRACTICAL NO. 3

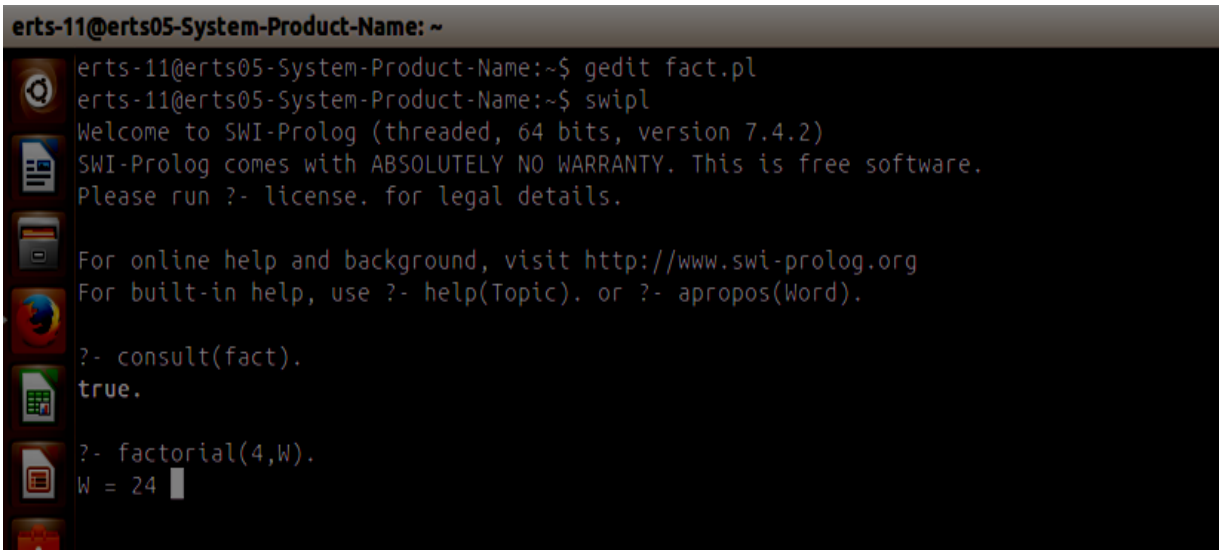
AIM: To implement some basic programs using Prolog Language.

1] FACTORIAL OF A NUMBER

PROGRAM:

```
factorial(0,1).
factorial(N,F) :-
 N>0,
 N1 is N-1,
 factorial(N1,F1),
 F is N * F1.
```

OUTPUT:

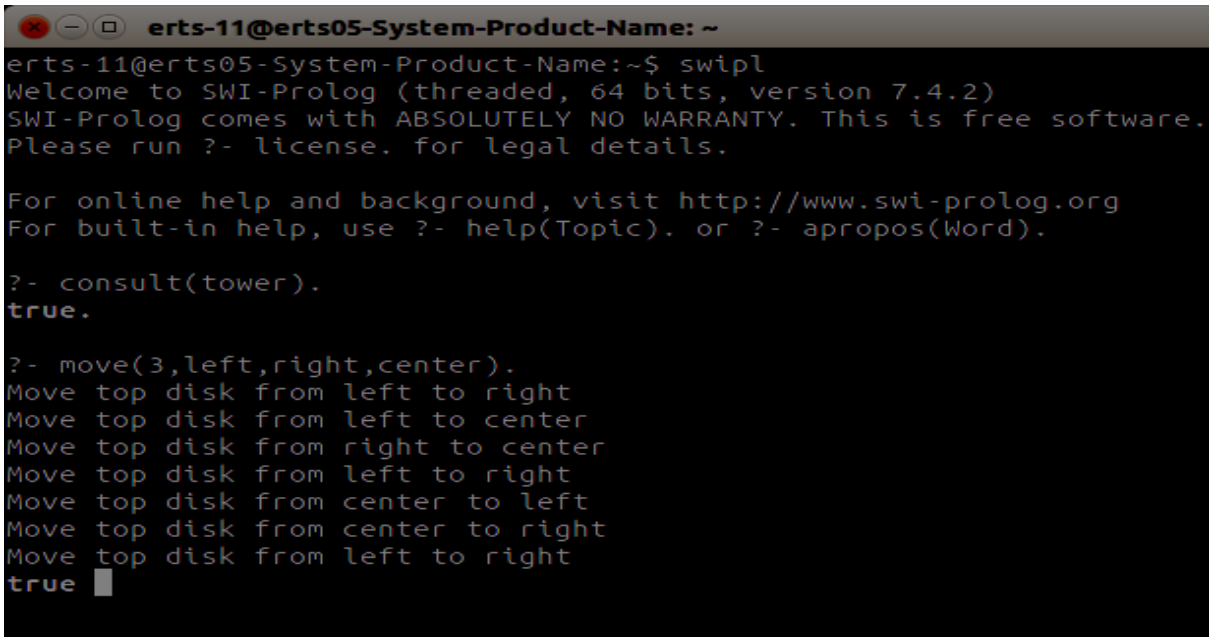


2] TOWER OF HANOI:

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:



3] AREA & CIRCUMFERENCE OF CIRCLE AND VOLUME OF A SPHERE:

PROGRAM:

```
circle:-
write('enter the radius of the circle :'), read(R),nl,
A is 3.14*R*R,
CF is 2*3.14*R,
V is 1.333333333*3.14*R*R*R,
write('Area of Circle is '), write(A), nl,
write('Circumference of Circle is '), write(CF),nl,
write('Volume of Sphere is '), write(V).
```

OUTPUT:

4] AREA OF RECTANGLE:

PROGRAM:

```
rectangle:-
write('enter the length :'), read(L),nl,
write('enter the breadth :'), read(B),nl,
A is L*B,
write('Area of Rectangle is '), write(A), nl.
```

OUTPUT:

```
erts-16@erts16: ~
erts-16@erts16:~$ swipl
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.6.4)
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

?- consult(rectangle).

true.

?- rectangle.
enter the length :5.

enter the breadth :4.

Area of Rectangle is 20
true.
```

5] FAMILY TREE:

PROGRAM:

```
:- discontiguous male/1, female/1, parent/2.
male(dicky).
male(randy).
male(mike).
male(don).
male(elmer).
female(anne).
female(rosie).
female(esther).
female(mildred).
female(greatgramma).
male(blair).
male(god).
female(god).
parent(don,randy).
parent(don,mike).
parent(don,anne).
parent(rosie,randy).
```

```
parent(rosie,mike).
parent(rosie,anne).
parent(elmer,don).
parent(mildred,don).
parent(esther,rosie).
parent(esther,dicky).
parent(greatgramma,esther).
parent(randy,blair).
male(mel).
male(teo).
parent(melsr,mel).
parent(melsr,teo).
american(anne).
american(X) :- ancestor(X,anne).
american(X) :- ancestor(anne,X).
relation(X,Y) :- ancestor(A,X), ancestor(A,Y).
father(X,Y) :- male(X),parent(X,Y).
father(god, _) :- male(god).
mother(X,Y) :- female(X),parent(X,Y).
son(X,Y) :- male(X),parent(Y,X).
daughter(X,Y) :- female(X),parent(Y,X).
grandfather(X,Y) :- male(X),parent(X,Somebody),parent(Somebody,Y).
aunt(X,Y) :- female(X),sister(X,Mom),mother(Mom,Y).
aunt(X,Y) :- female(X),sister(X,Dad),father(Dad,Y).
sister(X,Y) :- female(X),parent(Par,X),parent(Par,Y), X \= Y.
uncle(X,Y) :- brother(X,Par),parent(Par,Y).
cousin(X,Y) :- uncle(Unc , X),father(Unc,Y).
ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- parent(X,Somebody),ancestor(Somebody,Y).
brother(X,Y) :- male(X),parent(Somebody,X),parent(Somebody,Y), X \= Y.
```

OUTPUT:

```
erts-12@erts12:~$ gedit familytree.pl
erts-12@erts12:~$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 7.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult(familytree).
true.

?- parent(don,randy).
true .

?- parent(rosie,randy).
true.

?- brother(randy,mike).
true .

?- brother(anne,randy).
false.

?- sister(anne,randy).
true .

?- sister(anne,mike).
true .

?- brother(randy,anne).
true .

?- grandmother(elmer,randy).
ERROR: Undefined procedure: grandmother/2 (DWIM could not correct goal)
?- grandfather(elmer,Who).
Who = randy ;
Who = mike ;
Who = anne.

?- son(randy,don).
```

```
?- son(randy,don).
true .

?- daughter(rosie,Who).
Who = esther.

?- ancestor(greatgramma,Who).
Who = esther .

?- aunt(anne,blair).
true .

?- uncle(Who,blair).
Who = mike .

?- uncle(Who,blair).
Who = mike ;
Who = mike ;
false.

?- father(Who,randy)
|    .
Who = don .

?- mother(esther,Who).
Who = rosie .

?- mother(esther,Who).
Who = rosie ;
Who = dicky.

?- cousin(randy,Who).
false.

?- cousin(mike,randy).
false.

?- trace.
true.
```

```
?- trace.
true.

[trace]  ?- mother(esther,Who).
   Call: (8) mother(esther, _7976) ? creep
   Call: (9) female(esther) ? creep
   Exit: (9) female(esther) ? creep
   Call: (9) parent(esther, _7976) ? creep
   Exit: (9) parent(esther, rosie) ? creep
   Exit: (8) mother(esther, rosie) ? creep
Who = rosie .

[trace]  ?- mother(esther,Who).
   Call: (8) mother(esther, _7760) ? creep
   Call: (9) female(esther) ? creep
   Exit: (9) female(esther) ? creep
   Call: (9) parent(esther, _7760) ? creep
   Exit: (9) parent(esther, rosie) ? creep
   Exit: (8) mother(esther, rosie) ? creep
Who = rosie ;
   Redo: (9) parent(esther, _7760) ? creep
   Exit: (9) parent(esther, dicky) ? creep
   Exit: (8) mother(esther, dicky) ? creep
Who = dicky.

[trace]  ?- n
|
[1]+  Stopped                 swipl
erts-12@erts12:~$
```

CONCLUSION: Thus basic programs using Prolog Language has been successfully implemented.