PROGRAM:

PROGRAM:

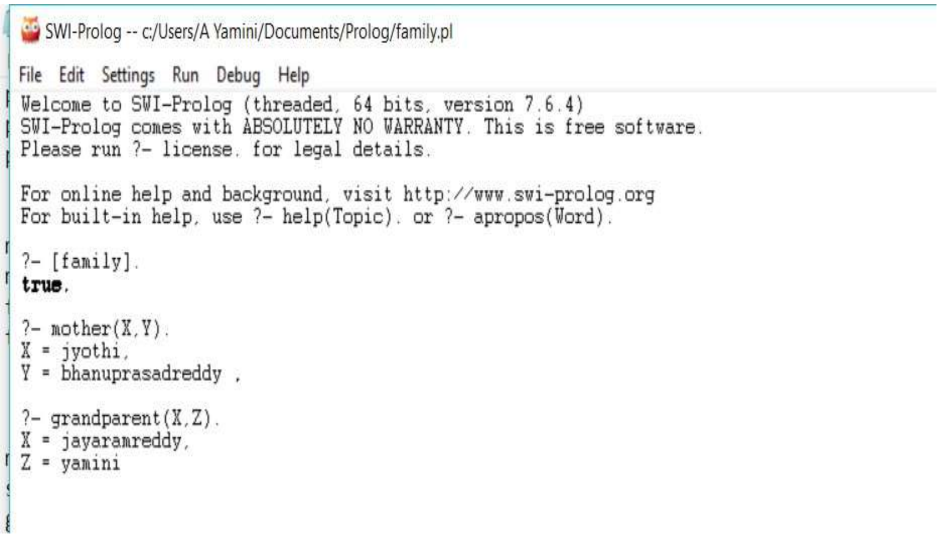
EX.NO: **DATE:**	**FAMILY TREE**

AIM:

ALGORITHM:

```
parent(jayaramreddy,bhanuprasadreddy).
parent(jyothi,bhanuprasadreddy).
parent(bhanuprasadreddy,yamini).
male(jayaramreddy). male(bhanuprasadreddy).
female(jyothi). female(yamini). mother(X,Y):-
parent(X,Y),female(X). sister(X,Y):-
parent(Z,X),parent(Z,Y),female(X).
grandparent(X,Z):-parent(X,Y),parent(Y,Z).
```

OUTPUT:


SWI-Prolog -- c:/Users/A Yamini/Documents/Prolog/family.pl
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [family].
true.

?- mother(X,Y).
X = jyothi,
Y = bhanuprasadreddy .

?- grandparent(X,Z).
X = jayaramreddy,
Z = yamini
```

SAVEETHA SCHOOL OF ENGINEERING DEPARTMENT OF CSE RUBRICS			
SLNO	**INDEX**	**MAXIMUM MARKS**	**MARKS AWARDED**
1.	**AIM**		
2.	**ALGORITHM**		
3.	**PROGRAM**		
4.	**OUTPUT**		
5.	**RESULT**		
6.	**OBSERVATION**		
7.	**VIVA**		
8	**TOTAL**		
SIGNATURE			

RESULT:

PROGRAM:

EX.NO: **DATE:**	**FACTORIAL**

AIM:

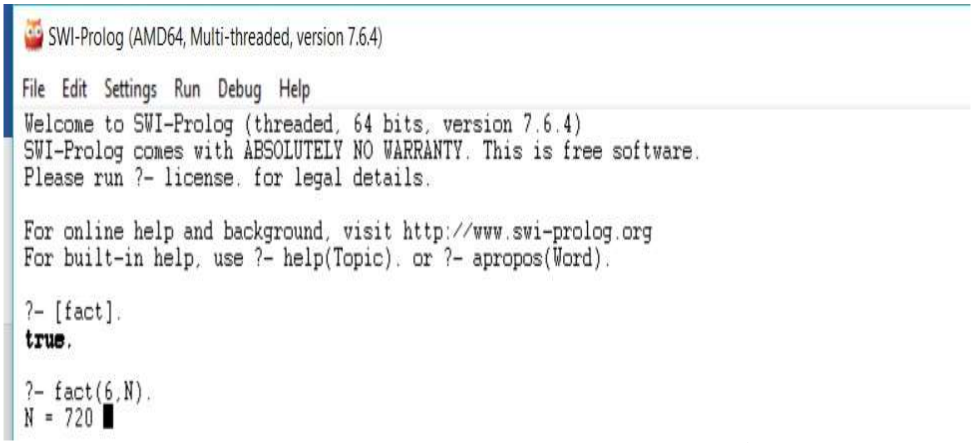
ALGORITHM:

```
fact(0, 1). fact(N,
F) :-
(% The below is for +ve factorialN > 0->
(N1 is N - 1, fact(N1,F1),
F is N * F1);
```

% The below is for -ve factorial

N<0->(N1 is N+1,fact(N1, F1),F is N * F1)).

OUTPUT:


SWI-Prolog (AMD64, Multi-threaded, version 7.6.4)
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [fact].
true.

?- fact(6,N).
N = 720
```

SAVEETHA SCHOOL OF ENGINEERING DEPARTMENT OF CSE RUBRICS			
SLNO	**INDEX**	**MAXIMUM MARKS**	**MARKS AWARDED**
1.	**AIM**		
2.	**ALGORITHM**		
3.	**PROGRAM**		
4.	**OUTPUT**		
5.	**RESULT**		
6.	**OBSERVATION**		
7.	**VIVA**		
8	**TOTAL**		
SIGNATURE			

RESULT:

PROGRAM:

GCD OF TWO NUMBERS

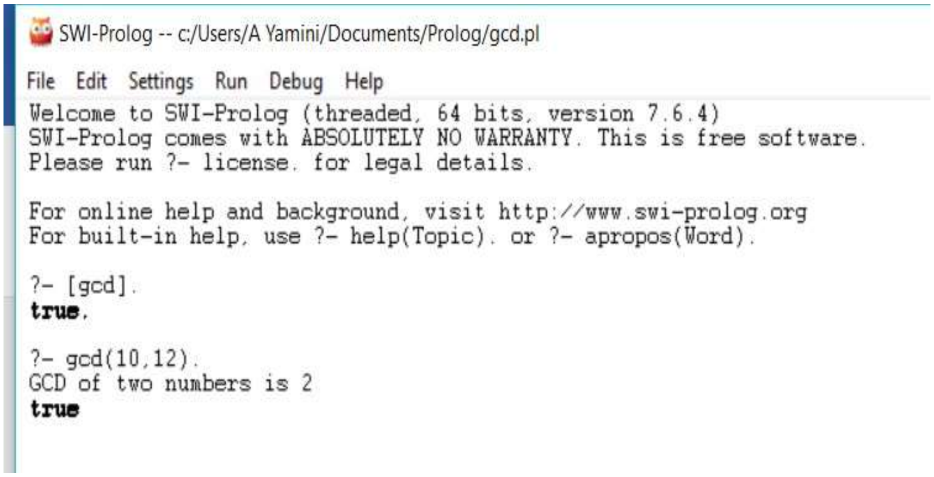
AIM:

ALGORITHM:

```
gcd(X,Y):-X=Y,write('GCD of two numbers is '),write(X);
 X=0,write('GCD of two numbers is '),write(Y);
Y=0,write('GCD of two numbers is '),write(X);
 Y>X,Y1 is Y-X,gcd(X,Y1);
 X>Y,Y1 is X-Y,gcd(Y1,Y).
```

EX.NO: **DATE:**	

OUTPUT:


SWI-Prolog -- c:/Users/A Yamini/Documents/Prolog/gcd.pl
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [gcd].
true.

?- gcd(10,12).
GCD of two numbers is 2
true
```

SAVEETHA SCHOOL OF ENGINEERING DEPARTMENT OF CSE RUBRICS			
SLNO	**INDEX**	**MAXIMUM MARKS**	**MARKS AWARDED**
1.	**AIM**		
2.	**ALGORITHM**		
3.	**PROGRAM**		
4.	**OUTPUT**		
5.	**RESULT**		
6.	**OBSERVATION**		
7.	**VIVA**		
8	**TOTAL**		
SIGNATURE			

RESULT:

PROGRAM:

INPUT FROM USER

AIM:

ALGORITHM:

PROGRAM: reference("yamini",

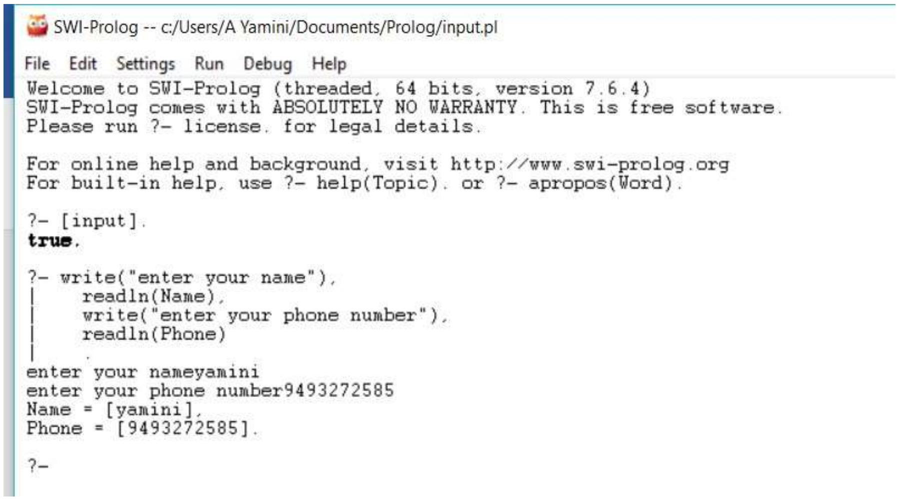
"9493272585"). reference("radhika",

"8919666297"). reference("hemanth",

"9642499090"). reference("jayaram

reddy", "9490013093").

OUTPUT:



SWI-Prolog -- c:/Users/A Yamini/Documents/Prolog/input.pl
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [input].
true.

?- write("enter your name"),
|    readln(Name),
|    write("enter your phone number"),
|    readln(Phone)
|    .
enter your nameyamini
enter your phone number9493272585
Name = [yamini],
Phone = [9493272585].

?-
```

SAVEETHA SCHOOL OF ENGINEERING DEPARTMENT OF CSE RUBRICS			
SLNO	**INDEX**	**MAXIMUM MARKS**	**MARKS AWARDED**
1.	**AIM**		
2.	**ALGORITHM**		
3.	**PROGRAM**		
4.	**OUTPUT**		
5.	**RESULT**		
6.	**OBSERVATION**		
7.	**VIVA**		
8	**TOTAL**		
SIGNATURE			

RESULT:

OUTPUT FROM USER

AIM:

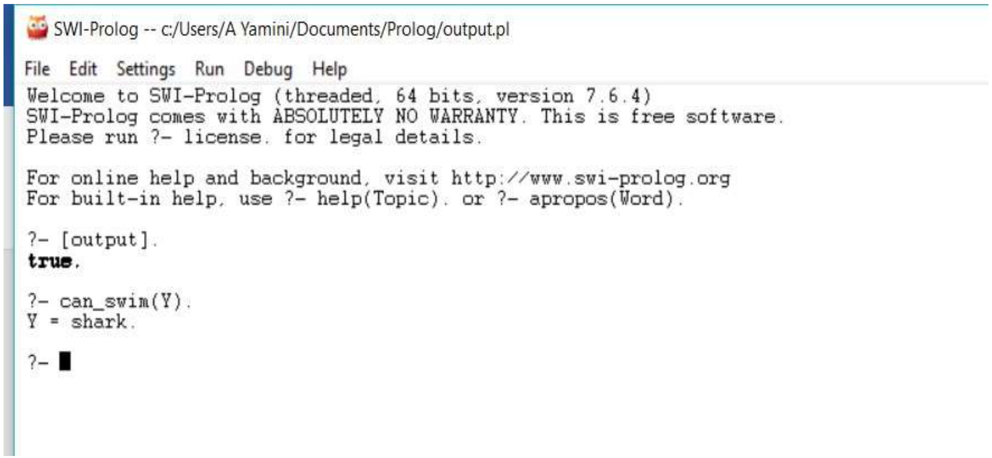
ALGORITHM:

PROGRAM: type(ungulate,animal).

EX.NO: **DATE:**	

```
type(fish,animal).
is_a(zebra,ungulate).
is_a(herring,fish). is_a(shark,fish).
lives(zebra,on_land). lives(frog,on_land). lives(frog,in_water).
lives(shark,in_water). can_swim(Y):- type(X, animal),
is_a(Y,X), lives(Y, in_water).
```

OUTPUT:


SWI-Prolog -- c:/Users/A Yamini/Documents/Prolog/output.pl
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [output].
true.

?- can_swim(Y).
Y = shark.

?- █
```

SAVEETHA SCHOOL OF ENGINEERING
DEPARTMENT OF CSE RUBRICS

SLNO	INDEX	MAXIMUM MARKS	MARKS AWARDED
1.	AIM		
2.	ALGORITHM		
3.	PROGRAM		
4.	OUTPUT		
5.	RESULT		
6.	OBSERVATION		
7.	VIVA		

8	TOTAL		
SIGNATURE			

RESULT:

EX.NO: 12 **DATE:**	**MONKEY BANANA**

AIM:

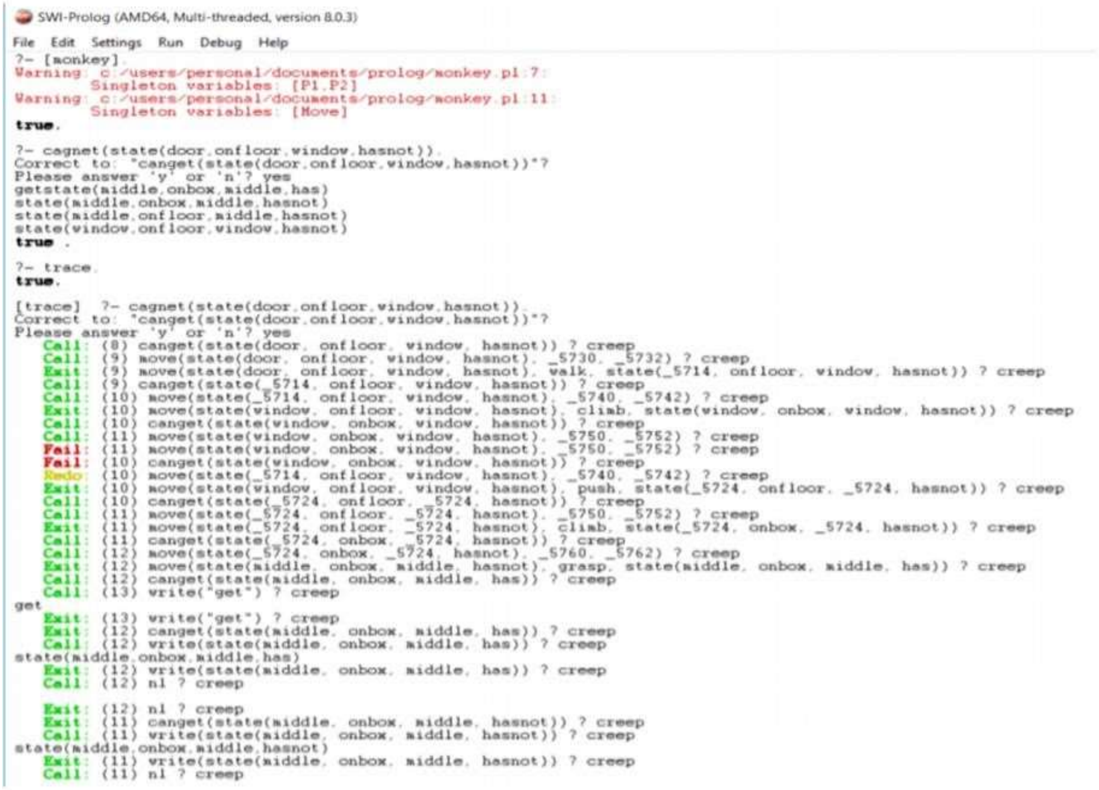
ALGORITHM:

PROGRAM:

```
move(state(middle,onbox,middle,hasnot),
grasp,state(middle,onbox,middle,has)).
```

```
move(state(P,onfloor,P,hasnot),climb,
state(P,onbox,P,hasnot)).
move(state(P,onfloor,P,hasnot),push,
state(P1,onfloor,P1,hasnot)).
move(state(P1,onfloor,B,hasnot),walk,
state(P2,onfloor,B,hasnot)).
canget(state(_,_,_,h
as)) :-
write("get").
canget(State1) :-
move(State1,Move,
State2),
canget(State2),
write(State2),nl.
```

OUTPUT:


SWI-Prolog (AMD64, Multi-threaded, version 8.0.3)
File Edit Settings Run Debug Help

```
?- [monkey].
Warning: c:/users/personal/documents/prolog/monkey.pl:7:
         Singleton variables: [P1,P2]
Warning: c:/users/personal/documents/prolog/monkey.pl:11:
         Singleton variables: [Move]
true.

?- cagnet(state(door,onfloor,window,hasnot)).
Correct to: "canget(state(door,onfloor,window,hasnot))"?
Please answer 'y' or 'n'? yes
getstate(middle,onbox,middle,has)
state(middle,onbox,middle,hasnot)
state(middle,onfloor,middle,hasnot)
state(window,onfloor,window,hasnot)
true .

?- trace.
true.

[trace]  ?- cagnet(state(door,onfloor,window,hasnot)).
Correct to: "canget(state(door,onfloor,window,hasnot))"?
Please answer 'y' or 'n'? yes
   Call: (8) canget(state(door, onfloor, window, hasnot)) ? creep
   Call: (9) move(state(door, onfloor, window, hasnot), _5730, _5732) ? creep
   Exit: (9) move(state(door, onfloor, window, hasnot), walk, state(_5714, onfloor, window, hasnot)) ? creep
   Call: (9) canget(state(_5714, onfloor, window, hasnot)) ? creep
   Call: (10) move(state(_5714, onfloor, window, hasnot), _5740, _5742) ? creep
   Exit: (10) move(state(window, onfloor, window, hasnot), climb, state(window, onbox, window, hasnot)) ? creep
   Call: (10) canget(state(window, onbox, window, hasnot)) ? creep
   Call: (11) move(state(window, onbox, window, hasnot), _5750, _5752) ? creep
   Fail: (11) move(state(window, onbox, window, hasnot), _5750, _5752) ? creep
   Fail: (10) canget(state(window, onbox, window, hasnot)) ? creep
   Redo: (10) move(state(_5714, onfloor, window, hasnot), _5740, _5742) ? creep
   Exit: (10) move(state(window, onfloor, window, hasnot), push, state(_5724, onfloor, _5724, hasnot)) ? creep
   Call: (10) canget(state(_5724, onfloor, _5724, hasnot)) ? creep
   Call: (11) move(state(_5724, onfloor, _5724, hasnot), _5750, _5752) ? creep
   Exit: (11) move(state(_5724, onfloor, _5724, hasnot), climb, state(_5724, onbox, _5724, hasnot)) ? creep
   Call: (11) canget(state(_5724, onbox, _5724, hasnot)) ? creep
   Call: (12) move(state(_5724, onbox, _5724, hasnot), _5760, _5762) ? creep
   Exit: (12) move(state(middle, onbox, middle, hasnot), grasp, state(middle, onbox, middle, has)) ? creep
   Call: (12) canget(state(middle, onbox, middle, has)) ? creep
   Call: (13) write("get") ? creep
get
   Exit: (13) write("get") ? creep
   Exit: (12) canget(state(middle, onbox, middle, has)) ? creep
   Call: (12) write(state(middle, onbox, middle, has)) ? creep
state(middle,onbox,middle,has)
   Exit: (12) write(state(middle, onbox, middle, has)) ? creep
   Call: (12) nl ? creep

   Exit: (12) nl ? creep
   Exit: (11) canget(state(middle, onbox, middle, hasnot)) ? creep
   Call: (11) write(state(middle, onbox, middle, hasnot)) ? creep
state(middle,onbox,middle,hasnot)
   Exit: (11) write(state(middle, onbox, middle, hasnot)) ? creep
   Call: (11) nl ? creep
```

SAVEETHA SCHOOL OF ENGINEERING DEPARTMENT OF CSE RUBRICS			
SLNO	INDEX	MAXIMUM MARKS	MARKS AWARDED
1.	AIM		
2.	ALGORITHM		
3.	PROGRAM		
4.	OUTPUT		
5.	RESULT		
6.	OBSERVATION		
7.	VIVA		
8	TOTAL		
SIGNATURE			

RESULT:

EX.NO: **DATE:**	**LIST**

AIM:

ALGORITHM:

PROGRAM:

11 A) PRINT LIST:

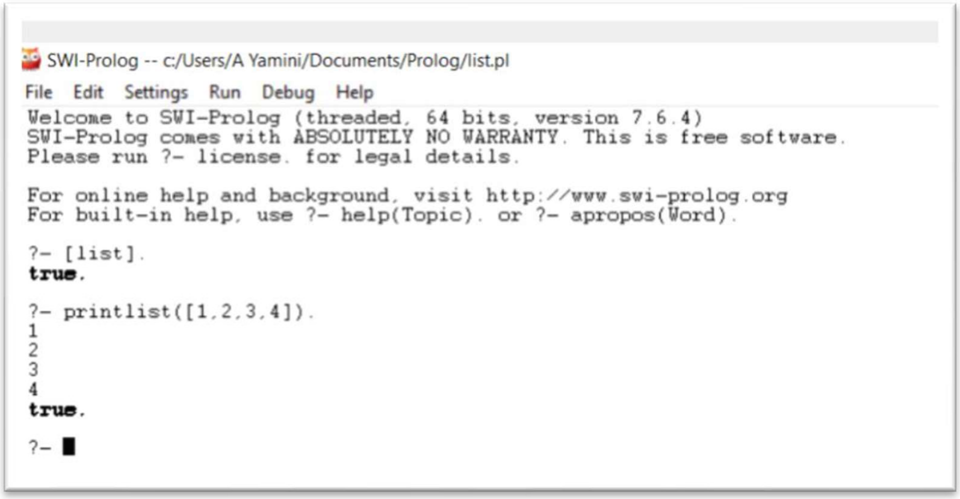
printlist([]).

printlist([X|List])

:- write(X),nl,

printlist(List).

OUTPUT:



SWI-Prolog -- c:/Users/A Yamini/Documents/Prolog/list.pl
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [list].
true.

?- printlist([1,2,3,4]).
1
2
3
4
true.

?- 
```

11 B) MEMBER IS PRESENT OR NOT IN A LIST

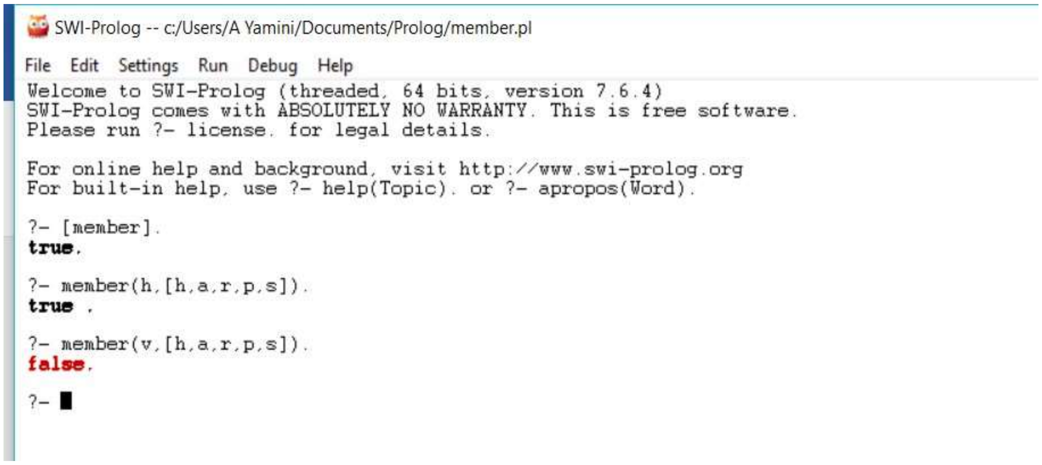
PROGRAM: member(X,List):- delete(X,List,_).

delete(X,[X|Tail],Tail).

delete(X,[Y|Tail1],[Y|Tail2]):-

delete(X,Tail1,Tail2).

OUTPUT:



SWI-Prolog -- c:/Users/A Yamini/Documents/Prolog/member.pl
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [member].
true.

?- member(h,[h,a,r,p,s]).
true .

?- member(v,[h,a,r,p,s]).
false.

?-
```

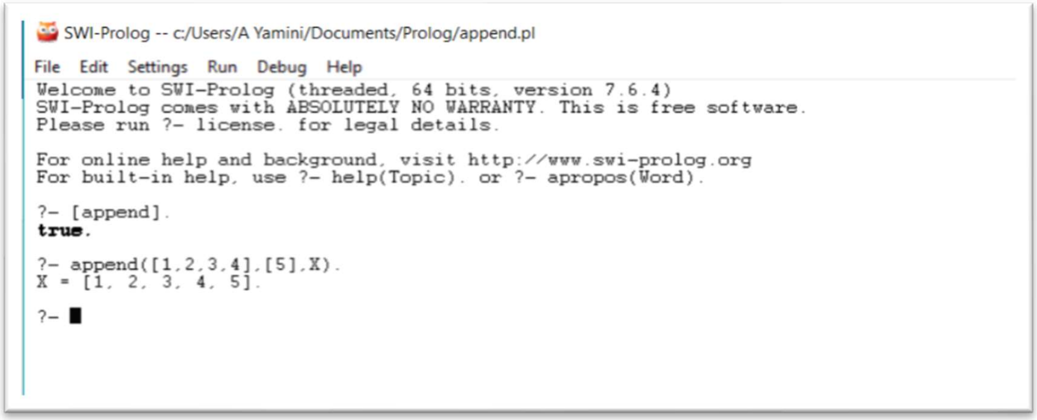
11 C) APPEND LIST:

```
append([],L,L).

append([X|L1],L2,[X|L3]) :-

append(L1,L2,L3).
```

OUTPUT:



SWI-Prolog -- c:/Users/A Yamini/Documents/Prolog/append.pl
File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [append].
true.

?- append([1,2,3,4],[5],X).
X = [1, 2, 3, 4, 5].

?-
```

SAVEETHA SCHOOL OF ENGINEERING DEPARTMENT OF CSE RUBRICS			
SLNO	**INDEX**	**MAXIMUM MARKS**	**MARKS AWARDED**
1.	**AIM**		
2.	**ALGORITHM**		
3.	**PROGRAM**		
4.	**OUTPUT**		
5.	**RESULT**		
6.	**OBSERVATION**		
7.	**VIVA**		
8	**TOTAL**		

SIGNATURE	

RESULT:

EX.NO: 13 **DATE:**	**MEDICAL DIAGNOSIS**

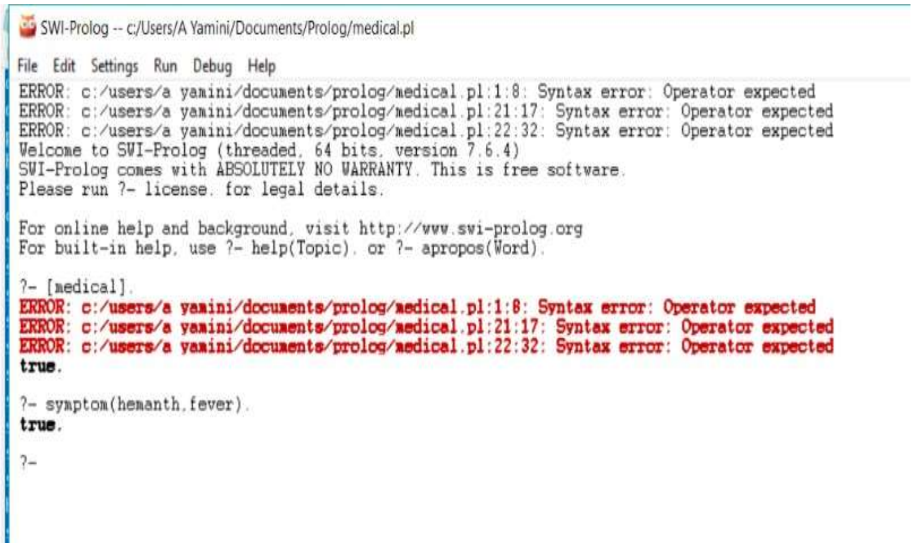
AIM:

ALGORITHM:

PROGRAM:

```
domains
disease,indication,name=
symbol predicates
hypothesis(name,disease)
symptom(name,indicatio
n) clauses
symptom(yamini,fever).
symptom(yamini,rash) .
symptom(yamini,headach
e).
symptom(yamini,runn_n
ose).
symptom(hemanth,chills)
.
symptom(hemanth,fever).
symptom(hemnth,headac
he).
symptom(radhika,runny_
nose).
symptom(radhika,rash).
symptom(radhika,flu).
hypothesis(Patient,mease
ls):
symptom(Patient,fever),
symptom(Patient,cough),
symptom(Patient,conjunc
tivitis),
symptom(Patient,r ash).
hypothesis(Patient,germa
n_measl es):
symptom(Patient,f ev er ),
symptom(Patient,headache),
symptom(Patient,runny_nose),
symptom(Patient,rash).
```

OUTPUT:



SWI-Prolog -- c:/Users/A Yamini/Documents/Prolog/medical.pl
File Edit Settings Run Debug Help

```
ERROR: c:/users/a yamini/documents/prolog/medical.pl:1:8: Syntax error: Operator expected
ERROR: c:/users/a yamini/documents/prolog/medical.pl:21:17: Syntax error: Operator expected
ERROR: c:/users/a yamini/documents/prolog/medical.pl:22:32: Syntax error: Operator expected
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
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).

?- [medical].
ERROR: c:/users/a yamini/documents/prolog/medical.pl:1:8: Syntax error: Operator expected
ERROR: c:/users/a yamini/documents/prolog/medical.pl:21:17: Syntax error: Operator expected
ERROR: c:/users/a yamini/documents/prolog/medical.pl:22:32: Syntax error: Operator expected
true.

?- symptom(hemanth,fever).
true.

?-
```