SWI-PROLOG Report

1. Write a program to find the sum of two numbers.

```
?-
% c:/Users/Acer/Documents/Kiran AI SwiProlog/abc.pl compiled 0.00 sec, 1 clauses
?- sum().
Enter first number : 69.
Enter second number : |: 9.
69 + 9 = 78
true.

?-
```

abc.pl

File Edit Browse Compile Prolog Pce Help

abc.pl

```
sum:-
write('Enter first number : '),read(X),
write('Enter second number : '),read(Y),
Z is X + Y,
write(X),write(' + '),write(Y),write(' = '),write(Z).
```

sum/0: (not loaded) unreferenced Line: 1

2. Write the program to find the family relation.

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog/FamilyRelation.pl compiled 0.00 sec, 16 clauses
?- father('Asha Kaji',Child).
Child = 'Anita' .

?- mother(Parent,'Sanjay').
Parent = 'Chaitya Maya'.

?- sister('Anita','Sanjay').
true.

?- grandfather('Purna','Sanjay').
true.

?- husband('Asha Kaji','Chaitya Maya').
false.

?- granddaughter('Anita','Purna').
true.

?- male(Person)
|
Person = 'Asha Kaji' .

?- husband('Asha Kaji',Spouse).
Spouse = 'Chitya Maya'.

?-
```

FamilyRelation.pl

File Edit Browse Compile Prolog Pce Help

FamilyRelation.pl

```
male('Asha Kaji').
female('Chaitya Maya').
female('Anita').
male('Sanjay').
father('Asha Kaji','Anita').
father('Asha Kaji','Sanjay').
mother('Chaitya Maya','Anita').
mother('Chaitya Maya','Sanjay').
husband('Asha Kaji','Chitya Maya').
sister('Anita','Sanjay').
brother('Sanjay','Anita').
grandfather('Purna','Sanjay').
grandfather('Purna','Anita').
father('Purna','Asha Kaji').
grandson('Sanjay','Purna').
granddaughter('Anita','Purna').
```

user:male/1: (loaded) 2 facts

3. Write the program to find the medical diagnosis.

```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.8)
File  Edit  Settings  Run  Debug  Help
?- husband('Asha Kaji',Spouse).
Spouse = 'Chitya Maya'.

?-
% c:/Users/Acer/Documents/Kiran AI SwiProlog/Medical Diagonosis.p
?- diagnose.
Welcome to the Medical Diagnosis System!
Please answer the following questions:
Do you have fever? (yes/no): no.
Do you have cough? (yes/no): |: yes.
Do you have sore_throat? (yes/no): |: no.
Do you have fatigue? (yes/no): |: no.
Do you have sneezing? (yes/no): |: yes.
Do you have runny_nose? (yes/no): |: no.
Do you have shortness_of_breath? (yes/no): |: yes.
Do you have chills? (yes/no): |: no.
Do you have sweating? (yes/no): |: yes.
Do you have headache? (yes/no): |: no.
Do you have frequent_urination? (yes/no): |: no.
Do you have increased_thirst? (yes/no): |: yes.
Do you have weight_loss? (yes/no): |: no.
Sorry, no diagnosis could be determined. Please consult a doctor.
true.

?- symptom(fever),disease(flu).
false.

?- symptom(fever),disease(heart attack).
ERROR: Syntax error: Operator expected
ERROR: symptom(fever),disease(heart
ERROR: ** here **
ERROR:  attack) .
?-
```

```
Medical Diagonosis.pl
File  Edit  Browse  Compile  Prolog  Pce  Help
Medical Diagonosis.pl
% Facts: Symptoms for each disease
disease(flu) :- symptom(fever), symptom(cough), symptom(sore_throat), symptom(fatigue).
disease(cold) :- symptom(cough), symptom(sneezing), symptom(runny_nose).
disease(covid19) :- symptom(fever), symptom(cough), symptom(shortness_of_breath), symptom(fatigue)
disease(malaria) :- symptom(fever), symptom(chills), symptom(sweating), symptom(headache).
disease(diabetes) :- symptom(frequent_urination), symptom(increased_thirst), symptom(weight_loss),
ymptom(fatigue).

% User input for symptoms
ask_symptom(Symptom) :-
    format('Do you have ~w? (yes/no): ', [Symptom]),
    read(Response),
    (Response == yes -> assertz(symptom(Symptom)); true).

% Diagnosis
diagnose :-
    write('Welcome to the Medical Diagnosis System!'), nl,
    write('Please answer the following questions:'), nl,
    % List all possible symptoms to check
    ask_symptom(fever),
    ask_symptom(cough),
    ask_symptom(sore_throat),
    ask_symptom(fatigue),
    ask_symptom(sneezing),
    ask_symptom(runny_nose),
    ask_symptom(shortness_of_breath),
    ask_symptom(chills),
    ask_symptom(sweating),
    ask_symptom(headache),
    ask_symptom(frequent_urination),
    ask_symptom(increased_thirst),
    ask_symptom(weight_loss),
    % Check for diseases
    (   disease(Disease)
    ->  format('You may have ~w. Please consult a doctor for confirmation.', [Disease])
    ;   write('Sorry, no diagnosis could be determined. Please consult a doctor.')
    ),
    % Clear the knowledge base for symptoms
    retractall(symptom(_)).
```

4. Write a program to find Area, Perimeter, Circle area and Circumference.

```
?-
% c:/Users/Acer/Documents/Kiran AI SwiProlo
?- area().
Enter the length of the rectangle : 6
|: .
Enter the breadth of the rectangle : |: 5.
Area of the rectangle is 30
true.

?- perimeter().
Enter the length of the rectangle : 6.
Enter the breadth of the rectangle : |: 5.
Perimeter of the rectangle is 22
true.

?- circlearea().
Enter the radius of the circle : 3.
The area of the circle is 28.285714285714285
true.

?- circumference().
Enter the radius of the circle :7.
The circumference of the circle is 44.0
true.

?-
```

```
area.pl
File  Edit  Browse  Compile  Prolog  Pce  Help
area.pl
area:-
write('Enter the length of the rectangle : '),read(L),
write('Enter the breadth of the rectangle : '),read(B),
A is L * B,
write('Area of the rectangle is '),write(A).

perimeter:-
write('Enter the length of the rectangle : '),read(X),
write('Enter the breadth of the rectangle : '),read(Y),
P is 2 * (X + Y),
write('Perimeter of the rectangle is '),write(P).

circlearea:-
write('Enter the radius of the circle : '),read(R),
B is 22/7 * R * R,
write('The area of the circle is '),write(B).

circumference:-
write('Enter the radius of the circle :'),read(S),
C is 2 * 22 / 7 * S,
write('The circumference of the circle is '),write(C).
```

5. Arithmetic Operations using SWI Prolog

```
?-
% c:/Users/Acer/Documents/Kiran AI
?- operation().
Enter first number : 20
|: .
Enter second number : |: 30.
Sum is : 50
Difference is : -10
Multiplication is : 600
Division is : 0.6666666666666666
Integer division is : 0
true.

?- square().
Enter a number : 2
|: .
Square of 2 is 4
true.

?- cube().
Enter a number : 3.
Cube of 3 is 27
true.

?- modulus().
Enter first number : 2.
Enter second number : |: 3.
2 modulus 3 is 2
true.

?-
```

arithmeticOperations.pl

```
operation:-
write('Enter first number : '),read(X),
write('Enter second number : '),read(Y),
A is X + Y,
S is X - Y,
M is X * Y,
D is X / Y,
E is X//Y,
write('Sum is : '),write(A),nl,
write('Difference is : '),write(S),nl,
write('Multiplication is : '),write(M),nl,
write('Division is : '),write(D),nl,
write('Integer division is : '),write(E).

square:-
write('Enter a number : '),read(N),
Z is N * N,
write('Square of '),write(N),write(' is '),write(Z).

cube:-
write('Enter a number : '),read(M),
C is M * M * M,
write('Cube of '),write(M),write(' is '),write(C).

modulus:-
write('Enter first number : '),read(P),
write('Enter second number : '),read(Q),
R is P mod Q,
write(P),write(' modulus '),write(Q),write(' is '),write(R).
```

6. WAP to find the factorial of given Number.

```
?-
% c:/Users/Acer/Documents/Kiran AI SwiProlog
?- factorial(5,6).
false.

?- factorial(3,6).
true
```

factorial.pl

File Edit Browse Compile

factorial.pl

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

7. WAP to find the fibonacci equivalent number of given numbers

```
?-
% c:/Users/Acer/Documents/Kiran AI
?- fibo(5,X).
X = 8
```

fibonacci.pl

File Edit Browse

fibonacci.pl

```
fibo(0,1).
fibo(1,1).
fibo(N,X):-
N>1,
N1 is N-1,
N2 is N-2,
fibo(N1,F1),
fibo(N2,F2),
X is F1+F2.
```

8. WAP to find GCD of two numbers.

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog/gcd.pl compiled 0.00 sec, 3 clauses
?- gcd(48,18,D).
D = 6 .

?- gcd(48,18,6).
true
```

gcd.pl

File Edit Browse Compile Prolog Pce Help

gcd.pl

```
gcd(X,X,X).
gcd(X,Y,D):-
X<Y,
Y1 is Y-X,
gcd(X,Y1,D).
gcd(X,Y,D):-
Y<X,
gcd(Y,X,D).
```

9. WAP to find the solution of tower of hanoi problem.

```
?-
% c:/Users/Acer/Documents/Kiran AI
?- move(3,a,b,c).
Move top disk from a to b
Move top disk from a to c
Move top disk from b to c
Move top disk from a to b
Move top disk from c to a
Move top disk from c to b
Move top disk from a to b
true
```

toh.pl

```
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).
```

10. WAP to find the LCM of given number.

```
?-
% c:/Users/Acer/Documents/Kiran AI SwiProlog/lcm.pl compiled 0.00 sec, 0 clauses
?-
|    lcm(12,15,L).
L = 60
```

lcm.pl

File Edit Browse Compile Prolog Pce Help

lcm.pl

```
gcd(X,X,X).
gcd(X,Y,D):-
X<Y,
Y1 is Y-X,
gcd(X,Y1,D).
gcd(X,Y,D):-
Y<X,
gcd(Y,X,D).
lcm(X,Y,L):-
gcd(X,Y,D),
L is X*Y//D.
```

11. WAP to find cube of a number.

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog/Cube.pl compiled 0.00 sec, 1 clauses
?- cube().
Enter a number : 3
|: .
Cube of 3 is 27
true.

?-
```

Cube.pl

File Edit Browse Compile Prolog Pce Help

Cube.pl

```
cube:-
write('Enter a number : '),
read(X),
Y is X * X * X,
write('Cube of '),write(X),write(' is '),write(Y).
```

12. WAP to concatenate two lists.

```
?-
% c:/Users/Acer/Documents/Kiran AI
?-
|    conc([1,2],[3,4],Result).
Result = [1, 2, 3, 4].

?-
```

Concat.pl

```
conc([],L,L).
conc([X|L1],L2,[X|L3]):-
conc(L1,L2,L3).
```

13. WAP to remove the first occurrence of an element X from a list.

```
% c:/Users/Acer/Documents/Kiran AI
?-
|    delt(3,[1,2,3,4,3],Result).
Result = [1, 2, 4, 3] .

?-
```

Remove.pl

```
delt(X,[X|Tail],Tail).
delt(X,[Y|Tail],[Y|Tail1]):-
delt(X,Tail,Tail1).
```

14. WAP to calculate the length of a given list.

```
% c:/Users/Acer/Documents/Kiran AI
?-
|    list_length([a,b,c,d],L).
L = 4.

?-
```

Listlength.pl

```
list_length([],0).
list_length([_|T],L):-list_length(T,L1),L is L1+1.
```

15. WAP to check if an element X is a member of a given list.

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog
?-
|    member(3,[1,2,3,4]).
true
```

List.pl

```
member(X,[X|Tail]).
member(X,[Head|Tail]):-
member(X,Tail).
```

16. WAP to reverse a list.

```
% c:/Users/Acer/Documents/Kiran AI
.

?- list_reverse([1,2,3,4],L).
L = [4, 3, 2, 1].

?-
```

reverse.pl

```
list_reverse([],[]).
list_reverse([H|T],L):-
list_reverse(T,R),append(R,[H],L).
```

17. WAP to determine the greater of two numbers (X and Y) and binds it to Z.

```
% c:/Users/Acer/Documents/Kiran AI
?- greatest(5,8,Z).
Z = 8.

?-
```

greatest3.pl

```
greatest(X,A,B,C):-
A=<B,B=<C,
X is C.
greatest(X,A,B,C):-
A>=B,A>=C,
X is A.
greatest(X,A,B,C):-
X is B.
```

18. WAP to for concatenate lists, reverse a list, and check if a list is a palindrome.

```
% c:/Users/Acer/Documents/Kiran AI
?-
|   pal([r,a,d,a,r]).
true.

?- pal([h,e,l,l,o]).
false.
```

palindrome.pl

```
concate([],L,L).
concate([X|L1],L2,[X|L3]):-
concate(L1,L2,L3).
reverse([],[]).
reverse([H|T],X):-
reverse(T,Y),
concate(Y,[H],X).
pal(X):-
reverse(X,X).
```

19. Write a Prolog program to generate all permutations of a given list.

```
% c:/Users/Acer/Documents/Kiran AI
?-
|   permutation([a,b,c],P).
P = [a, b, c]
```

permutation.pl

```
permutation([], []).

permutation(L, [X|P]) :-
    del(X, L, L1),
    permutation(L1, P).

del(X, [X|T], T).
del(X, [H|T], [H|R]) :-
    del(X, T, R).
```

20. Write a Prolog program to calculate the average of a given list of numbers.

```
% c:/Users/Acer/Documents/Kiran AI
.

?- main.
[8,4],6
true.

?-
```

average.pl

```
list([8,4]).
average_easy(List,Avg):-
sum_(List,Sum),
length_(List,Length),
Avg is Sum/Length.
sum_([],0).
sum_([H|T],Sum):-
sum_(T,Temp),
Sum is Temp + H.
length_([],0).
length_([_|B],L):-
length_(B,Ln),
L is Ln+1.
main:-
list(X),
average_easy(X,Ans),
writeln((X,Ans)).
```

21. Write a Prolog program to compute all factors of a given number N.

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog/factor.pl compiled 0.00 sec, 3 clauses
?-
|    factor(28,L).
L = [1, 2, 4, 7, 14, 28].

?-
```

factor.pl

File Edit Browse Compile Prolog Pce Help

factor.pl

```
factor(N, L) :-
        factor(N, 1, [], L).

factor(N, X, LC, L) :-
        0 is N mod X,
        !,
        Q is N / X,
        (Q = X ->
            sort([Q | LC], L)
        ;
            (Q > X ->
               X1 is X+1,
               factor(N, X1, [X, Q|LC], L)
            ;
              sort(LC, L)
            )
        ).

factor(N, X, LC, L) :-
        Q is N / X,
        (Q > X ->
            X1 is X+1,
            factor(N, X1, LC, L)
        ;
            sort(LC, L)
        ).
```

22. Write a program to find the relation. (Which locations are in Asia ?)

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog/9.2.pl compiled 0.00 sec, 9 clauses
?- located_in(X,kathmandu).
X = nccs .

?- located_in(X,nepal).
X = nccs
```

9.2.pl

File Edit Browse Compile Prolog Pce Help

9.2.pl

```
located_in(nccs, kathmandu).
located_in(thamel, kathmandu).
located_in(dharahara, kathmandu).
located_in(pokhara, kaski).
located_in(X, nepal) :- located_in(X, kathmandu).
located_in(X, nepal) :- located_in(X, lalitpur).
located_in(X, western_region) :- located_in(X, kaski).
located_in(X, asia) :- located_in(X, nepal).
located_in(X, asia) :- located_in(X, western_region).
```

23. Write a program to find the greatest number among given number.

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog/
?- greatest(X,55,66,77).
X = 77
```

greatest3.pl

File Edit Browse Compile

greatest3.pl

```
greatest(X,A,B,C):-
A=<B,B=<C,
X is C.
greatest(X,A,B,C):-
A>=B,A>=C,
X is A.
greatest(X,A,B,C):-
X is B.
```

24. What are the values of S, M, E, N, D, O, R, and Y that satisfy the equation SEND + MORE = MONEY?

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog/crypt.pl compiled 0.00 sec, 0 clauses
.

?- solve(S,M,E,N,D,O,R,Y).
S = 9,
M = 1,
E = 5,
N = 0,
D = 6,
O = 8,
R = 7,
Y = 2
```

crypt.pl

File Edit Browse Compile Prolog Pce Help

crypt.pl

```
value(1). value(2). value(3).
value(4). value(5). value(6).
value(7). value(8). value(9). value(0). %% Define digit values
solve(S,M,E,O,N,R,D,Y) :-
 M is 1, %% Our constraints first
 O is 0,
 value(S),
 S >= 8,
 value(Y), %% Each var must be a value
 value(D),
 value(R),
 value(N),
 value(E),
 S \= M, S\=E, S\=O, S\=N, S\=R, S\=D, S\=Y, %% Ensure uniqueness
 M\=E, M\=O, M\=N, M\=R, M\=D, M\=Y,
 E\=O, E\=N, E\=R, E\=D, E\=Y,
 O\=N, O\=R, O\=D, O\=Y,
 N\=R, N\=D, N\=Y,
 R\=D, R\=Y,
 D\=Y,
 Y is (D+E) mod 10, %% check if SEND+MORE
 C1 is truncate((D+E)/10), %% = MONEY
 E is (C1+N+R) mod 10, %% could use // instead
 C2 is truncate((C1+N+R)/10), %% of truncate
 N is (C2+E+O) mod 10,
 C3 is truncate((C2+E+O)/10),
 O is (C3+S+M) mod 10.
```

25. Write a program to print a square of stars with a side length X.

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog/star.pl compiled
.

?- square(5).
*****
*****
*****
*****
*****
true
```

star.pl

File Edit Browse Compile Prolog Pce Help

star.pl

```
square(1) :-
    write('*****'), nl.
square(X) :-
    X > 1,
    write('*****'), nl,
    X1 is X - 1,
    square(X1).
```

26. Write a program to find the path from the start node to the goal node using breadth-first search.

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog/bfs.pl compiled 0.00 sec, 15 clauses
.

?- find_path(a,g,Path).
Path = [g, d, a] .

?- find_path(a,l,Path).
Path = [l, f, c, a]
```

bfs.pl

File Edit Browse Compile Prolog Pce Help

bfs.pl

```
goal(g). % Define the goal node.
arc(a, b).
arc(a, c).
arc(a, d).
arc(c, k).
arc(c, f).
arc(d, g).
arc(d, h).
arc(d, i).
arc(f, l).
arc(h, m).

% Breadth-First Search (BFS)
bf([[Start|Path]|_], Start, [Start|Path]). % Found the goal.
bf([[K|Path]|Paths], Goal, Solution) :-
    K \== Goal,
    bagof([M, K | Path], (arc(K, M), \+ member(M, [K | Path])), New_Paths),
    append(Paths, New_Paths, Paths1),
    !,
    bf(Paths1, Goal, Solution);
    bf(Paths, Goal, Solution). % Fallback if no new paths are found.
bf([], _, []). % If there are no paths left to explore.

% Wrapper predicate to initiate BFS
find_path(Start, Goal, Solution) :-
    bf([[Start]], Goal, Solution).
```

user:goal/1: (loaded) 1 fact Line: 1

27. Write a program to find the path from the start node to the goal node using depth-first search.

```
% c:/Users/Acer/Documents/Kiran AI SwiProlog
?-
|    solve(a,Solution).
Solution = [j, e, b, a] .

?- solve(c,Solution).
Solution = [f, c]
```

dfs.pl

```
% Graph representation
s(a, b).
s(a, c).
s(b, d).
s(b, e).
s(c, f).
s(c, g).
s(d, h).
s(e, i).
s(e, j).
s(f, k).
% Goal nodes
goal(f).
goal(j).
% Check if an element is a member of a list
member(X, [X|_]).
member(X, [_|Tail]) :- member(X, Tail).
% Solve using depth-first search
solve(Node, Solution) :-
    depthfirst([], Node, Solution).
% If the current node is a goal, return the solution
depthfirst(Path, Node, [Node|Path]) :-
    goal(Node).
% Explore the next node if the current one is not a goal
depthfirst(Path, Node, Solution) :-
    s(Node, Node1),
    not(member(Node1, Path)),
    depthfirst([Node|Path], Node1, Solution).
```