

ARTIFICIAL INTELLIGENCE

QUESTION-01 Write a prolog program to calculate the sum of two numbers.

Sum(X,Y,Z):- Z is X+Y.

```
Sum.pl
sum(X, Y, Z) :- Z is X+Y.
```

```
?-
% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Sum.pl compiled 0.00 sec, 1 clauses
?- sum(4,5,Z).
Z = 4+5.

?-
% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Sum.pl compiled 0.00 sec, -2 clauses
?- sum(4,5,Z).
Z = 9.

?- sum(10,45,Z).
Z = 55.

?-
```

QUESTION-02 Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y.

max(X,Y,M):-X>Y, M is X.

max(X,Y,M):-Y>=X, M is Y.

```
Maximum.pl
max(X, Y, M) :- X>Y, M is X.
max(X, Y, M) :- Y>=X, M is Y.
```

```
% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Maximum.pl compiled 0.00 sec, 2 clauses
?- max(8,10,Z).
Z = 10.

?- max(18,10,Z).
Z = 18.

?- max(18,18,Z).
Z = 18.
```

QUESTION-03 Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N

fact(0,1).

fact(N,X):-N1 is N-1,fact(N1,Y),X is Y*N,!.

Factorial.pl

fact(0,1) .

fact(N,X):-N1 is N-1, fact(N1,Y), X is Y*N, !.

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Factorial.pl compiled 0.00 sec, 2 clauses

?- fact(10,F).

F = 3628800.

?- fact(5,F).

F = 120.

?-

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Factorial.pl compiled 0.00 sec, 0 clauses

?- fact(5,F).

F = 120 .

?- fact(5,F).

F = 120 .

?- fact(4,F).

F = 24 .

?- fact(4,F).

F = 24 ;

ERROR: Stack limit (1.0Gb) exceeded

ERROR: Stack sizes: local: 0.9Gb, global: 77.8Mb, trail: 0Kb

ERROR: Stack depth: 10,192,716, last-call: 0%, Choice points: 3

ERROR: In:

ERROR: [10,192,716] user:fact(-10192702, _20393666)

ERROR: [10,192,715] user:fact(-10192701, _20393686)

ERROR: [10,192,714] user:fact(-10192700, _20393706)

ERROR: [10,192,713] user:fact(-10192699, _20393726)

ERROR: [10,192,712] user:fact(-10192698, _20393746)

ERROR:

ERROR: Use the --stack_limit=size[KMG] command line option or

ERROR: ?- set_prolog_flag(stack_limit, 2_147_483_648). to double the limit.

?- |

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Factorial.pl compiled 0.00 sec, 0 clauses

?- | fact(4,F).

F = 24.

?- fact(4,F).

F = 24.

Question-04 Write a program in PROLOG to implement generate fib(N,T) where T represents the Nth term of the Fibonacci series.

fab(1,0).

fab(2,1).

fab(N,X):-N1 is N-1, N2 is N-2, fab(N1,X1), fab(N2,X2), X is X1+X2, !.

Fabonacci.pl [modified]

fab(1,0) .

fab(2,1) .

fab(N,X):-N1 is N-1, N2 is N-2, fab(N1,X1), fab(N2,X2), X is X1+X2, !.

?-

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Fabonacci.pl compiled 0.00 sec, 3 clauses

?- fab(10,X).

X = 34.

?-

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Fabonacci.pl compiled 0.00 sec, 0 clauses

?- fab(10,X).

X = 34.

Question-05 Write a Prolog program to implement GCD of two numbers.

gcd(0,A,A):-!.

gcd(A,0,A):-!.

gcd(A,B,R):- B1 is mod(A,B),gcd(B,B1,R).

```
GCD.pl
gcd(0,A,A):-!.
gcd(A,0,A):-!.
gcd(A,B,R):- B1 is mod(A,B),gcd(B,B1,R).

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/GCD.pl compiled 0.00 sec, -1 clauses
?- gcd(10,2,R).
R = 2.

?-
% c:/users/anjali yadav/onedrive/documents/prolog/fabonacci compiled 0.00 sec, 0 clauses
% c:/users/anjali yadav/onedrive/documents/prolog/gcd compiled 0.00 sec, 0 clauses
?- gcd(10,2,R).
R = 2.

?- gcd(20,80,R).
R = 20.
```

Question-06 Write a Prolog program to implement power (Num,Pow, Ans) : where Num is raised to the power Pow to get Ans.

pow(X,0):-!.

pow(Num,Pow, Ans):- Ans is Num^Pow.

```
Power.pl
pow(X,0):-!.
pow(Num,Pow, Ans):- Ans is Num^Pow.

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Power.pl compiled 0.00 sec, 1 clauses
?- pow(2,5,A).
A = 32.
```

Question-07 Prolog program to implement multi (N1, N2, R) : where N1 and N2 denotes the numbers to be multiplied and R represents the result

multi(X,0).

multi(N1, N2,R):-R is N1*N2.

```
Multi.pl
multi(X,0).
multi(N1,N2,R):-R is N1*N2.

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Multi.pl compiled 0.00 sec, 1 clauses
?- multi(10,2,R).
R = 20.
```

Question -08 Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.

member(X,[X|Tail]).

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

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

?- member(1, [1,22,22,2,2,2]) .
true .

?- member(5, [1,22,22,2,2,2]) .
false.
```

Question_09 Write a Prolog program to implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.

conc([],L2,L2).

conc([H|L1],L2,[H|L3]):-conc(L1,L2,L3).

```
Concat.pl
conc([], L2, L2) .
conc([H|L1], L2, [H|L3]) :- conc(L1, L2, L3) .

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Concat.pl compiled 0.00 sec, 0 clauses
?- conc([1,2,3,4],[a,b],L).
L = [1, 2, 3, 4, a, b].
```

Question-10 Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.

conc([],L2,L2).

conc([H|L1],L2,[H|L3]):-conc(L1,L2,L3).reverse([],[]).

reverse([H|Tail],R):-reverse(Tail,RevTail),conc(RevTail,[H],R).

```
reverse.pl
conc([], L2, L2) .
conc([H|L1], L2, [H|L3]) :- conc(L1, L2, L3) .reverse([], []).
reverse([H|Tail], R) :- reverse(Tail, RevTail), conc(RevTail, [H], R) .

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/reverse.pl compiled 0.00 sec, 0 clauses
?- reverse([1,2,3,4,5,6],A).
A = [6, 5, 4, 3, 2, 1].
```

Question-11 Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not.

conc([],L2,L2).

conc([H|L1],L2,[H|L3]):-conc(L1,L2,L3).

palindrome([]).

palindrome([_]).

palindrome(L):-conc([H|T],[H],L),palindrome(T).

```
Palidrome.pl
conc([],L2,L2).
conc([H|L1],L2,[H|L3]):-conc(L1,L2,L3).
palindrome([]).
palindrome([_]).
palindrome(L):-conc([H|T],[H],L),palindrome(T).

ERROR:   format:~s: type error: text expected, found uri( C:/Users/anjali yadav/OneDrive/document
% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Palidrome.pl compiled 0.00 sec, 2 clauses
?- palindrome([]).
true.

?- palindrome([1]).
true.

?- palindrome([1,2,3]).
false.

?- palindrome([1,2,1]).
true.

?- palindrome([1,2,1]).
true;
false.

?- palindrome([1]).
true;
false.
```

Question-12 Write a Prolog program to implement sumlist(L, S) so that S is the sum of a given list L

sum([],0).

sum([H|T],S):-sum(T,ST), S is H+ST.

```
SumList.pl
sum([],0).
sum([H|T],S):-sum(T,ST), S is H+ST.

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/SumList.pl compiled 0.00 sec, 0 clauses
?- sum([1,2,3,4],S).
S = 10.
```

Question-13 Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.

evenlength([]).

evenlength([_|T]):-oddlength(T).

oddlength([_]).

oddlength([_|T]):-evenlength(T).

OddEven.pl

```
evenlength([ ]).
evenlength([_|T]):-oddlength(T).
oddlength([ ]).
oddlength([_|T]):-evenlength(T).
```

```
% c:/users/anjali yadav/onedrive/documents/prolog/oddeven compiled 0.00 sec, 1 clauses
?- evenlength([1,2,2,3,3]).
false.

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

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

?- oddlength([1,2,3,4,2,4]).
false.
```

Question-14 . Write a Prolog program to implement nth_element (N, L, X) where N is the desired position, L is a list and X represents the Nth element of L.

nth_element(1,[H|_],H).

nth_element(N,[_|T],X):-N1 is N-1,nth_element(N1,T,X).

nElement.pl

```
nth_element(1,[H|_],H).
nth_element(N,[_|T],X):-N1 is N-1,nth_element(N1,T,X).
```

```
% c:/users/anjali yadav/onedrive/documents/prolog/nElement.pl compiled 0.00 sec, 0 clauses
?- nth_element(1,[1,2,3,1],X).
X = 1.

?- nth_element(5,[1,2,3,1],X).
false.

?- nth_element(2,[1,2,3,1],X).
X = 2.
```

Question-15 Write a Prolog program to implement maxlist(L, M) so that M is the maximum number in the list

max(X,Y,Z):-X>Y,Z is X.

max(X,Y,Z):-Y>=X , Z is Y.

max_list([H|T],R):-max_list(T,R1),max(H,R1,R).

Max_list.pl

```
max(X,Y,Z):-X>Y,Z is X.  
max(X,Y,Z):- Y>=X , Z is Y.  
  
max_list([H|T],R):-max_list(T,R1),max(H,R1,R) .
```



```
ERROR: NO PERMISSION TO LOAD THE IMPORTED PROCEDURE lists:MAX_LIST/2  
% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/Max_list.pl compiled 0.00 sec, 2 clauses  
?- max_list([1,2,3,4,4,5,20],M).  
M = 20.  
  
?- max_list([1,2,3,4,4,5,2],M).  
M = 5.
```

Question-16 Write a prolog program to implement insert_nth (I, N, L, R) that inserts an item I into Nth position of list L to generate a list R.

insertn(I,1,List,[I|List]).

insertn(I,Pos,[H|List],[H|R]):-Pos1 is Pos-1, insertn(I,Pos1,List,R).

insert_at_nth.pl

```
insertn(I,1,List,[I|List]).  
insertn(I,Pos,[H|List],[H|R]):-Pos1 is Pos-1, insertn(I,Pos1,List,R) .
```

```
?-  
% c:/users/anjali yadav/onedrive/documents/prolog/insert_at_nth compiled 0.00 sec, 0 clauses  
?- insertn([2],3,[9,8,7,6,5,4,3,2,1],R).  
R = [9, 8, [2], 7, 6, 5, 4, 3, 2|...] .  
  
?- insertn([2,3],[9,8,7,6,5,4,3,2,1],R).  
ERROR: Syntax error: Illegal start of term  
ERROR: insertn([2,3,[9,8,7,6,5,4,3,2,1],  
ERROR: ** here **  
ERROR: R)  
?- insertn([A,3],[9,8,7,6,5,4,3,2,1],R).  
ERROR: Syntax error: Illegal start of term  
ERROR: insertn([A,3,[9,8,7,6,5,4,3,2,1],  
ERROR: ** here **  
ERROR: R)  
?- insertn([A],3,[9,8,7,6,5,4,3,2,1],R).  
R = [9, 8, [A], 7, 6, 5, 4, 3, 2|...] .  
  
?- insertn([2],3,[9,8,7,6,5,4,3,2,1],R).  
R = [9, 8, [2], 7, 6, 5, 4, 3, 2|...] .  
  
?- insertn(2,3,[9,8,7,6,5,4,3,2,1],R).  
R = [9, 8, 2, 7, 6, 5, 4, 3, 2|...] .  
  
?- ■
```

Question-17 Write a Prolog program to implement delete_nth (N, L, R) that removes the element on Nth position from a list L to generate a list R.

remove([_|T],1,T).

remove([H|T],Pos,[H|Result]):-Pos1 is Pos-1, remove(T,Pos1,Result).

removeN.pl

```
remove([_|T],1,T).  
remove([H|T],Pos,[H|Result]):-Pos1 is Pos-1, remove(T,Pos1,Result) .
```

```

% c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/removeN.pl compiled 0.00 sec, 2 clauses
?- remove([1,2,3,3,4,5,1],1,R).
R = [2, 3, 3, 4, 5, 1] .

?- remove([1,2,3,3,4,5,1],3,R).
R = [1, 2, 3, 4, 5, 1] .

```

Question-18 Write a program in PROLOG to implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list

merge(X,[],X).

merge([],Y,Y).

merge([X|X1],[Y|Y1],[X|Z]):-X<Y,!,merge(X1,[Y|Y1],Z).

merge([X|X1],[Y|Y1],[X,Y|Z]):-X=Y,!,merge(X1,Y1,Z).

merge([X|X1],[Y|Y1],[Y|Z]):-X>Y,!,merge([X|X1],Y1,Z).

merge_list.pl
<pre> merge(X, [], X) . merge([], Y, Y) . merge([X X1], [Y Y1], [X Z]) :- X<Y, !, merge(X1, [Y Y1], Z) . merge([X X1], [Y Y1], [X,Y Z]) :- X=Y, !, merge(X1, Y1, Z) . merge([X X1], [Y Y1], [Y Z]) :- X>Y, !, merge([X X1], Y1, Z) . % c:/Users/Anjali Yadav/OneDrive/Documents/Prolog/merge_list.pl compiled 0.00 sec, 0 clauses ?- merge([], [1,2], R). R = [1, 2]. ?- merge([1,2], [], R). R = [1, 2]. ?- merge([1,2], [5,6,1,2,3], R). R = [1, 2, 5, 6, 1, 2, 3]. </pre>