

ARTIFICIAL INTELLIGENCE PRACTICAL FILE

Name – MUHEB FAZLI

Exam Roll No -19020570049

Roll No - 2019/1457

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

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

Q1_Sum.pl

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

```
?- sum(1,2,Z).  
Z = 3.
```

```
?- sum(10,2,Z).  
Z = 12.
```

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.

Q2_Maximum.pl

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

```
?- max(1,2,M).  
M = 2.
```

```
?- max(11,2,M).  
M = 11.
```

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,!.

Q2_Maximum.pl Q3_Factorial.pl

```
fact(0,1) .  
fact(N,X):-N1 is N-1, fact(N1,Y), X is Y*N, !.
```

```
?- fact(0,F) .  
F = 1 .
```

```
?- fact(2,F) .  
F = 2 .
```

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, !.

Q4_Fabonacci.pl

```
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, !.
```

```
?- fab(3,X) .  
X = 1 .
```

```
?- fab(4,X) .  
X = 2 .
```

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).

Q5_GCD.pl

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

```
?- gcd(5,15,X).  
X = 5.
```

```
?- gcd(25,15,X).  
X = 5.
```

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.

Q6_Power.pl

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

```
?- pow(5,2,P).  
P = 25.
```

```
?- pow(10,3,P).  
P = 1000.
```

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.

Q7_Multi.pl

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

```
?- multi(5, 10, M).  
M = 50.
```

```
?- multi(25, 10, M).  
M = 250.
```

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).

Q8_Member.pl

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

```
?- member(5, [1, 2, 2, 3, 4, 5, 6, 6, 5]).  
true .
```

```
?- member([5], [1, 2, 2, 3, 4, 5, 6, 6, 5]).  
false.
```

```
?- member(15, [1, 2, 2, 3, 4, 5, 6, 6, 5]).  
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).

Q9_Concat.pl

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

```
?- conc([1,2,3,4,5],[2,4,'A'],M).
M = [1, 2, 3, 4, 5, 2, 4, 'A'].

?- conc([1,2,3,4,5],[2,4,4],M).
M = [1, 2, 3, 4, 5, 2, 4, 4].
```

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).
```

```
Q10_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).
```

```
?- reverse([1,2,3,4,5,6,7,8,9],R).
R = [9, 8, 7, 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).
```

Q11_Palindrome.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) .

?- palindrome([1]).
true .

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

Q12_SumList.pl

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

?- sum([1,23,4,5,5,6],S).
S = 44.

?- sum([1,25,6],S).
S = 32.
```

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).

Q13_OddEven.pl

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

```
?- evenlength([1,25,6,1]).  
true.
```

```
?- oddlength([1,25,6,1]).  
false.
```

```
?- oddlength([1,25,6]).  
true.
```

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).

Q14_nElement.pl

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

```
?- nth_element(10,[2,2,1,23,3],X).  
false.
```

```
?- nth_element(4,[2,2,1,23,3],X).  
X = 23.
```

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).

Q15_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).

?- max_list([-1,-2,-3,-12,-343,-2],Max).
Max = -1.
? ■
```

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).

Q16_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).

?- insertn(12,3,[1,2,3,4,5,6,7,8,9,0],M).
M = [1, 2, 12, 3, 4, 5, 6, 7, 8|...] .

?- insertn(12,0,[1,2,3,4,5,6,7,8,9,0],M).
false.

?- insertn(12,5,[1,2,3,4,5,6,7,8,9,0],M).
M = [1, 2, 3, 4, 12, 5, 6, 7, 8|...] .
```

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).

Q17_removeN.pl

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

```
?- remove([1,2,3,4,5,6,7,8,9,0],5,Result).
Result = [1, 2, 3, 4, 6, 7, 8, 9, 0] .

?- remove([1,2,3,4,5,6,7,8,9,0],0,Result).
false.

?- remove([1,2,3,4,5,6,7,8,9,0],20,Result).
false.
```

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).

Q18_Merge_list.pl

```
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([], [5,4,3,2,1], Result).
Result = [5, 4, 3, 2, 1].
```

```
?- merge([1,2,3,4,5], [], Result).
Result = [1, 2, 3, 4, 5].
```