

ARTIFICIAL INTELLIGENCE

ASSIGNMENT - 04

1. W.A.P.P to find factorial of a number.

```
%U19CS009
%BRIJESH ROHIT
fact(0,1).
fact(N,F):-
    N1 is N-1,
    fact(N1,F1),
    F is N*F1.
```

```
?- consult('u19cs009-ai-assign04-q1-factorial.pl').
true.

?- fact(10,ANS).
ANS = 3628800 .

?- fact(5,ANS).
ANS = 120 .

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

2. W.A.P.P to print Fibonacci series.

The Fibonacci sequence $f(1), f(2), f(3)...$ is:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55....

Example:

?- fib(6, R).

R = 8

```
%U19CS009
%BRIJESH ROHIT
fib(1,1).
fib(2,1).
fib(N,X):-
    N1 is N-1,
    N2 is N-2,
    fib(N1,X1),
    fib(N2,X2),
    X is X1+X2.
```

```
?- consult('u19cs009-ai-assign04-q2-fibonacci.pl').
true.

?- fib(1,ANS).
ANS = 1 .

?- fib(2,ANS).
ANS = 1 .

?-
|   fib(3,ANS).
ANS = 2 .

?- fib(10,ANS).
ANS = 55 .

?- fib(30,ANS).
ANS = 832040 .
```

3. W.A.P.P to finding the greatest common divider (GCD) and the least common multiple (LCM) of two integers.

```
%U19CS009
%BRIJESH ROHIT

lcm(A,B,R):-
    gcd(A,B,Res),
    R is A*B/Res.

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

calculate(A,B,GCD,LCM):-
    gcd(A,B,GCD),
    lcm(A,B,LCM).
```

```
?- consult('u19cs009-ai-assign04-q3-gcd-lcm.pl').
true.

?- calculate(17,25,GCD,LCM).
GCD = 1,
LCM = 425.

?- calculate(10,25,GCD,LCM).
GCD = 5,
LCM = 50.

?- calculate(91,63,GCD,LCM).
GCD = 7,
LCM = 819.
```

4. W.A.P.P.

A. To find length of the list.

```
%U19CS009
% BRIJESH ROHIT

%count number of elements in list
countEle([],0).
countEle([_|T],N):-
    countEle(T,N1),
    N is N1 + 1.
```

```
?- consult('u19cs009-ai-assign04-q4-lists.pl').
true.

?- %Length of List
|   countEle([1,2,3,4,5,6,7],Len).
Len = 7.

?- countEle([1,2,[3,4,5,6],7],Len).
Len = 4.

?- countEle([1,2,[3,4,5,6],[],[a,b,[c,[d]]],7],Len).
Len = 6.
```

B. To find first and last element of the list.

```
%first element of list
firstEle([]).
firstEle(X,[H|_]):-
    X is H.

%last element
lastEle(X,[X]).
lastEle(X,[_|L]):-
    lastEle(X,L).

firstAndLast([H|T]):-
    write("First element : "),
    firstEle(X,[H|T]),
    write(X),nl,
    write("Last element : "),
    lastEle(Y,[H|T]),
    write(Y).
```

```
?- consult('u19cs009-ai-assign04-q4-lists.pl').
true.

?- firstAndLast([1,2,3,4,5,6,7,8]).
First element : 1
Last element : 8
true .

?- firstAndLast([1,2,[1,2,3],a,v,c,[a,c,[b,d,[e]]]]).
First element : 1
Last element : [a,c,[b,d,[e]]]
true .

?- firstAndLast([1,2,[1,2,3]]).
First element : 1
Last element : [1,2,3]
true .
```

C. To find the nth element of the list.

```
%nth element in list
nthEle([],0):-
    write("Invalid Argument!!"),nl.
nthEle([H|_],1):-
    write("Nth element is : "),
    write(H).
nthEle([_|T],N):-
    N1 is N-1,
    nthEle(T,N1).
```

```
?- consult('u19cs009-ai-assign04-q4-lists.pl').
true.

?- nthEle([1,2,3,4,5,6,7,8,9],4).
Nth element is : 4
true .

?- nthEle([1,2,3,4,5,6,7,8,9],10).
false.

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

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

?- nthEle([1,2,3,4,5,6,7,8,9],3).
Nth element is : 3
true .

?- nthEle([1,2,[3,4,5,6,7],8,9],3).
Nth element is : [3,4,5,6,7]
true .
```

D. To increment each number in the list.

```
%increment each number in list
increment([],[]).
increment([H|T],[H1|T1]):-
    increment(T,T1),
    H1 is H+1.
```

```
?- consult('u19cs009-ai-assign04-q4-lists.pl').
true.

?- increment([1,2,3,4,5],NewList).
NewList = [2, 3, 4, 5, 6].

?- increment([1,2,3,4,5,0,5,2,4,2,3,10],NewList).
NewList = [2, 3, 4, 5, 6, 1, 6, 3, 5|...].
```

E. To reverse the list.

```
%add element at end
addAtEnd(H,[],[H]).
addAtEnd(X,[H|T1],[H|T2]):-
    addAtEnd(X,T1,T2).
```

```
%reverse a list
reverse([],[]).
reverse([H|T],Y):-
    reverse(T,T1),
    addAtEnd(H,T1,Y).
```

```
?- consult('u19cs009-ai-assign04-q4-lists.pl').
true.

?- reverse([1,2,3,4,5,6,7],Rev).
Rev = [7, 6, 5, 4, 3, 2, 1] .

?- reverse([1,2,3,4,[5,6,7]],Rev).
Rev = [[5, 6, 7], 4, 3, 2, 1] .

?- reverse([1,2,3,4,[5,6,7],a,b,c,[a,b,[c,d]]],Rev).
Rev = [[a, b, [c, d]], c, b, a, [5, 6, 7], 4, 3, 2, 1] .
```

F. To verify if a list has an even number of elements.

```
%check even or odd integer
check_even(N):-
    Y is N//2,Y*2==N
    ->format('Given List has \"EVEN\" number of elements.~n');
    format('Given List has \"ODD\" number of elements.~n').
%check if number of elements are even or odd
evenList([H|T]):-
    countEle([H|T],N), %from first ques
    check_even(N).
```

```
?- evenList([1,2,3,4,5,6,7]).
Given List has "ODD" number of elements.
true.

?- evenList([1,2,3,4,5,6,7,8]).
Given List has "EVEN" number of elements.
true.

?- evenList([1,2,3,4,[5,6,7,8]]).
Given List has "ODD" number of elements.
true.

?- evenList([1,2,3,[4,5,6,7,8]]).
Given List has "EVEN" number of elements.
true.
```

G. To count vowels in the list

```
%check vowels in list
vowels(X):-member(X,[a,e,i,o,u]).
countVowels([],0).
countVowels([H|T],N):-
    vowels(H),
    countVowels(T,N1),
    N is N1+1,
    !.
countVowels([_|T],N):-
    countVowels(T,N).
```

```
?- consult('u19cs009-ai-assign04-q4-lists.pl').
true.

?- countVowels([b,r,i,j,e,s,h,r,o,h,i,t],NumOfVowels).
NumOfVowels = 4.

?- countVowels([b,r,i,j,e,s,h,r,a,m,e,s,h,b,h,a,i,r,o,h,i,t],NumOfVowels).
NumOfVowels = 8.

?- countVowels([a,e,i,o,u],NumOfVowels).
NumOfVowels = 5.

?- countVowels([],NumOfVowels).
NumOfVowels = 0.

?- countVowels([b,r,i,j,e,s,h,[r,a,m,e,s,h,b,h,a,i],r,o,h,i,t],NumOfVowels).
NumOfVowels = 4.
```

H. To remove duplicates from the list.

```
%remove_duplicates
remove_duplicates([], []).

remove_duplicates([H | T], List) :-
    member(H, T),
    remove_duplicates(T, List).

remove_duplicates([H | T], [H|T1]) :-
    \+member(H, T),
    remove_duplicates(T, T1).
```

```
?- consult('u19cs009-ai-assign04-q4-lists.pl').
true.

?- remove_duplicates([b,r,i,j,e,s,h,r,a,m,e,s,h,b,h,a,i,r,o,h,i,t],NewList).
NewList = [j, m, e, s, b, a, r, o, h|...] .

?- remove_duplicates([b,r,i,j,e,s,h,r,o,h,i,t],NewList).
NewList = [b, j, e, s, r, o, h, i, t] .

?- remove_duplicates([a,b,a,b,a,a,a,b],NewList).
NewList = [a, b] .

?- remove_duplicates([a,a,a,a,a,a,a,a],NewList).
NewList = [a] .
```