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

Code:

```
fact(0,X):- X=1.
fact(M,X):- M>0, M1 is M-1,fact(M1,X1),X is X1*M .
```

```
?- [fact].
true.

?- fact(4, X).
X = 24 .

?- fact(9, X).
X = 362880 .

?-
```

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

Code:

```
fibo(1,Y):- Y is 1.
fibo(2,Y):- Y is 1.

fibo(X,Result):- X>2,X1 is X-1, X2 is X-2,fibo(X1,R1),fibo(X2,R2),Result is R1+R2.
```

```
?- [fibo].
true.

?- fibo(7, X).
X = 13 .

?- fibo(9, X).
X = 34 .

?- fibo(13, X).
X = 233 .

?-
```

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

Code:

```
%GCD
gcd(X,0,X).
gcd(X,Y,Z):-
```

```
R is mod(X,Y),
gcd(Y,R,Z).
%LCM
lcm(X,Y,LCM):-gcd(X,Y,GCD), LCM is X*Y//GCD.
```

```
?- [lcm].
true.

?- gcd(4, 20, X).
X = 4 .

?- lcm(7, 3, X).
X = 21 .

?-
```

W.A.P.P.
A. To find length of the list.
B. To find first and last element of the list.
C. To find the nth element of the list.
D. To increment each number in the list.
E. To reverse the list.
F. To verify if a list has an even number of elements.
G. To count vowels in the list
H. To remove duplicates from the list.

Code:

```
%finding length of the list
len([],0).
len([_|T], N) :-
len(T,N1),
N is N1 + 1.

%finding last element
last([H|[]],H).
last([_|T],L) :-
last(T,L).

%finding first element
first([H|_],H).

%finding find member of list
find(H,1,[H|_]).
find(X,N,[_|L]) :-
find(X,N1,L),
N is N1+1.
```

```
%incrementing each element by 1
inc([], []).
inc([H|T], [X|Y]) :-
inc(T,Y),
X is H+1.

%reversing list
rev([], Y, Y).
rev([H|T] , Y, R) :-
rev(T, [H|Y], R).

%if a list has an even number of elements.
odd([_|T]):-
even(T).
even([]).
even([_|T]) :-
odd(T).

%To count vowels in the list
vowel(X):- member(X,[a,e,i,o,u]).
countv([],0).
countv([X|T],N):- vowel(X),countv(T,N1), N is N1+1,!.
countv([X|T],N):- countv(T,N).

%removing duplicates
rem([], []).
rem([Head | Tail], Result) :-
member(Head, Tail), !,
rem(Tail, Result).
rem([Head | Tail], [Head | Result]) :-
rem(Tail, Result).
```

```
?- [string].
Warning: /home/maxmax/Desktop/u19cs019_sem6/Artificial_Intelligence/lab4/string.pl:43:
        Singleton variables: [X]
true.

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

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

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

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

?- inc([1, 2, 6, 4, 5],X).
X = [2, 3, 7, 5, 6].
```

```
?- rev([1, 2, 6, 4, 5], [],X).
X = [5, 4, 6, 2, 1].

?- even([1, 2, 6, 4, 5]).
false.

?- even([1, 2, 6, 4]).
true.

?- rem([1, 2, 6, 4, 5], X).
X = [1, 2, 6, 4, 5].

?- rem([1, 2, 2, 6, 4, 5], X).
X = [1, 2, 6, 4, 5].

?- countv([a, i, n, e, m, o, u, b],X).
X = 5.
```