TOPIC : MORE PROLOG PROGRAMS

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

```
sumlist([],0).

sumlist([H|T],S):-sumlist(T,S1),S is H+S1.
```

Write a Prolog program to implement two predicates evenlen(List) and oddlen(List) so that they are true if their argument is a list of even or odd length respectively

```
evelen([]).

evelen([_|[_|List]]):-evelen(List).

oddlen([_]).

oddlen([_|[_|List]]):-oddlen(List).
```

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|T],H).

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

Write a program in PROLOG to implement remove_dup (L, R) where L denotes the list with some duplicates and the list R denotes the list with duplicates removed.

```
member(X,[X|_]).

member(X,[_|Y]):- member(X,Y).

remove_dup(L,M):-dupacc(L,[],M).

dupacc([],A,A).
```

```
dupacc([H|T],A,L):-member(H,A),dupacc(T,A,L),!.

dupacc([H|T],A,L):-dupacc(T,[H|A],L).
```

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):- X=<Y,Z is Y.

maxlist([],0):-!.

maxlist([R],R):-!.

maxlist([H|T],R):-maxlist(T,R1),max(H,R1,R),!.
```

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(Item,List,1,[Item|List]).

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

Write a Program in PROLOG to implement sublist(S, L) that checks whether the list S is the sublist of list L or not. (Check for sequence or the part in the same order).

```
sublist([],[]).

sublist([First|Rest],[First|Sub]):- sublist(Rest,Sub).

sublist([_|Rest],Sub):-sublist(Rest,Sub).
```

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.

```
removen([_|List],1,List).
```

```
removen([H|List],Pos,[H|Result]):-Pos1 is Pos-1, removen(List,Pos1,Result).
```

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).
```

Write a PROLOG program that will take grammar rules in the following format:

NT □(NT | T)*
Where NT is any nonterminal, T is any terminal and Kleene star (*) signifies any number of repetitions, and generate the corresponding top-down parser, that is:
sentence □ noun-phrase, verb-phrase
determiner □ [the]
will generate the following:
sentence (I, O) :- noun-phrase(I,R), verb-phrase (R,O).
determiner ([the|X], X) :- !.

```
sentence(X,Y):-np(X,Z),vp(Z,Y).

np(X,Y):-det(X,Z),noun(Z,Y).

vp(X,Y):-verb(X,Z),np(Z,Y).

vp(X,Y):-verb(X,Y).

det([a|X],X).

det([an|X],X).

det([the|X],X).

noun([boy|X],X).

noun([girl|X],X).
```

```
noun([song|X],X).
noun([apple|X],X).
verb([sing|X],X).
verb([sings|X],X).
verb([eats|X],X).
```

Write a prolog program that implements Semantic Networks (ATN/RTN).

```
class(person).
class(male).
class(female).
class(height).
class(weight).
class(age).
isa(male,person).
isa(female,person).
isa(jane,female).
isa(john,male).
isa(david,male).
owns(john,house).
owns(jane,house).
owns(david,car).
hasprop(person,[height,weight,age]).
hasa(john,height,171).
hasa(john,weight,70).
```

```
hasa(john,age,31).

hasa(jane,height,154).

hasa(jane,weight,49).

hasa(jane,age,24).

hasa(david,height,167).

hasa(david,weight,69).

hasa(david,age,28).

isperson(X):-isa(X,person),!.

isperson(X):-isa(Y,person),isa(X,Y).

get_prop(X,[H|[]],P):-hasa(X,H,Y),append([],[Y],P).

get_prop(X,[H|T],P):-get_prop(X,T,P1),hasa(X,H,Y),append([Y],P1,P).

prop_of(X,P):-isperson(X),hasprop(person,P1),get_prop(X,P1,P),!.

hascar(X1):-isperson(X1),\+(class(X1)),owns(X1,car).
```

References/Resources

- Dan. W. Patterson, Artificial Intelligence and Expert Systems, Prentice Hall, 2004
- Elaine Rich, Kevin Knight, & Shivashankar B Nair, Artificial Intelligence, McGraw Hill, 3rd ed.,2009

NOTE: Please go through the above programs carefully and practice them (on machine if possible).