**Final Practical File**

**Name:** Ravish Ranjan

**Course:** BSc. Hons. Computer Science

**Semester:** 6th Semester **Paper:** Artificial Intelligence

**Roll no.:** 21058570040

**Que 1 WAP to check is a list is a subset of another.**

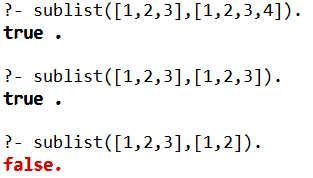
**Ans**

sublist(S,L):- append(L1,L2,L),append(S,L3,L2).

append([],L,L).

append([X|L1],L2,[X|L3]):-append(L1,L2,L3).

**#Output**

****

**Que 2 WAP to remove duplicates from a list.**

**Ans**

remove\_duplicates([],[]).

remove\_duplicates([H | T], List) :-

     member(H, T),

     remove\_duplicates( T, List).

remove\_duplicates([H | T], [H|T1]) :-

      not(member(H, T)),

      remove\_duplicates( T, T1).

**#Output**

****

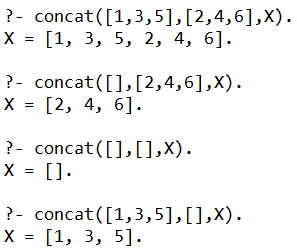
**Que 3 WAP to concatenate two lists.**

**Ans**

concat([],L,L).

concat([H|T],L,[H|TR]):- concat(T,L,TR).

**#Output**

****

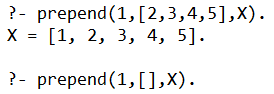
**Que 4 WAP to prepend an element in a list.**

**Ans**

prepend(I,[],[I]).

prepend(I,L,R):- concat([I],L,R).

**#Output**

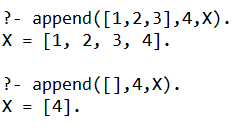
****

**Que 5 WAP to append an element in list.**

**Ans**

append(L,I,R):- concat(L,[I],R).

**#Output**

****

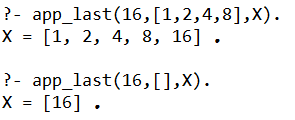
**Que 6 WAP to append an element in list using recursion.**

**Ans**

app\_last(I,[],[I]).

app\_last(I,[H|T],[H|TR]):- app\_last(I,T,TR).

**#Output**

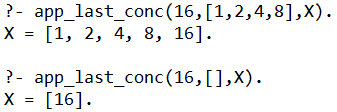
****

**Que 7 WAP to append an element in list using concatenation.**

**Ans**

app\_last\_conc(I,L,R):- concat(L,[I],R).

**#Output**

****

**Que 8 WAP to insert an element in list at nth position.**

**Ans**

insert(I,[],0,[I]).

insert(I,[H|T],AT,X):-

    AT > 0,

    AT=:=1 ->

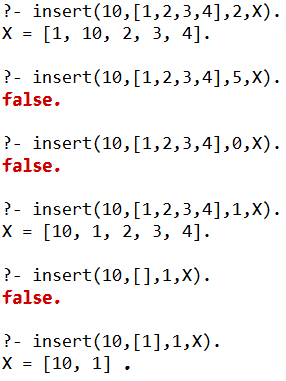
        prepend(H,T,R1),prepend(I,R1,X);

        AT1 is AT-1,

        insert(I,T,AT1,R2),

        prepend(H,R2,X).

**#Output**

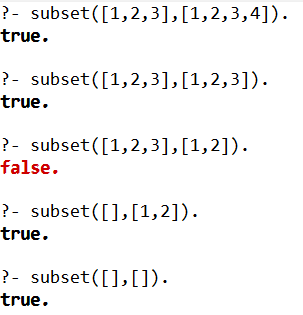
****

**Que 9 WAP to find if first list is a subset of second list using concat.**

**Ans**

subset(L,LR):- concat(\_,L2,LR),concat(L,\_,L2),!.

**#Output**

****

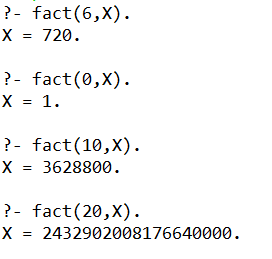
**Que 10 WAP to find factorial of a number.**

**Ans**

fact(0,1):- !   .

fact(X,R):- X>=0,X1 is X-1,fact(X1,R1),R is X\*R1.

**#Output**

****

**Que 11 WAP to get Fibonacci series’ nth element.**

**Ans**

generate\_fib(1,0).

generate\_fib(2,1).

generate\_fib(N,T):-

    N>2,

    N1 is N-1,

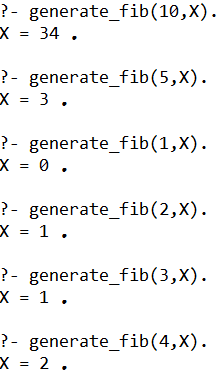
    N2 is N-2,

    generate\_fib(N2,T2),

    generate\_fib(N1,T1),

    T is T1+T2.

**#Output**

****

**Que 12 WAP to find nth element of a list.**

**Ans**

findnth([H],1,H).

findnth([H|T],AT,X):-

    AT >= 1,

    AT=:=1 ->

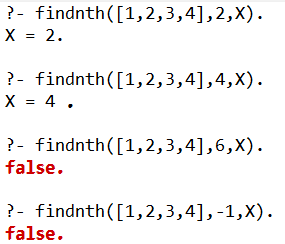
        X is H;

        AT1 is AT-1,

        findnth(T,AT1,X1),

        X is X1.

**#Output**

****

**Que 13 WAP to get greatest common division (GCD) of two numbers.**

**Ans**

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

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

gcd(X, Y, D) :-

    X > Y ->

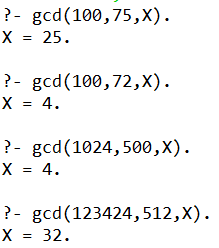
        T is X - Y,

        gcd(T, Y, D);

        T is Y - X,

        gcd(T,X,D).

**#Output**

****

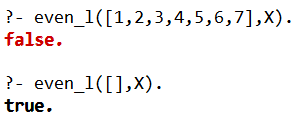
**Que 14 WAP to check if length of list is even.**

**Ans**

even\_l([],\_):- !.

even\_l([\_|T],R):- odd\_l(T,R).

**#Output**

****

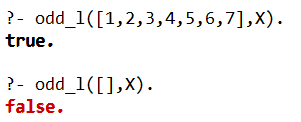
**Que 15 WAP to check if length of list is odd.**

**Ans**

odd\_l([\_],\_):- !.

odd\_l([\_|T],R):- even\_l(T,R).

**#Output**

****

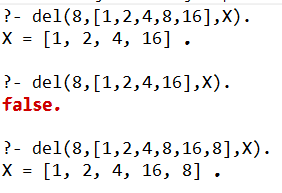
**Que 16 WAP to delete given element from list.**

**Ans**

del(X,[X|T],T).

del(X,[H|T],[H|T1]):- del(X,T,T1).

**#Output**

****

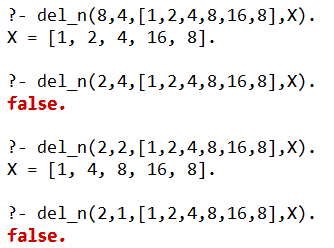
**Que 17 WAP to delete nth element from list.**

**Ans**

del\_n(X,1,[X|T],T):- !.

del\_n(X,N,[H|T],[H|R]):- N>1,N1 is N-1,del\_n(X,N1,T,R).

**#Output**

****

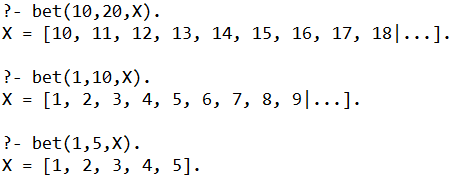
**Que 18 WAP to generate all integer between two numbers.**

**Ans**

bet(H,H,[H]):-!.

bet(L,H,[L|R]):- L<H,N is L+1,bet(N,H,R).

**#Output**

****

**Que 19 WAP to check if list is ordered in ascending order or not.**

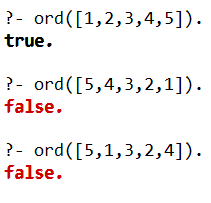
**Ans**

ord([]):-!.

ord([\_]):-!.

ord([A,B|T]):- A=<B,ord([B|T]).

**#Output**

****

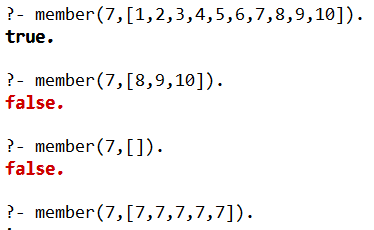
**Que 20 WAP to find a number in list.**

**Ans**

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

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

**#Output**

****

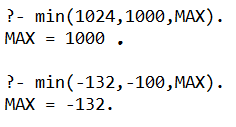
**Que 21 WAP to find minimum between two numbers.**

**Ans**

min(X,Y,MIN):-X>=Y,MIN is Y.

min(X,Y,MIN):- X<Y,MIN is X.

**#Output**

****

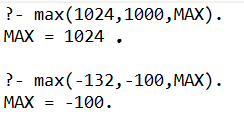
**Que 22 WAP to find maximum between two numbers.**

**Ans**

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

max(X,Y,MAX):- X<Y,MAX is Y.

**#Output**

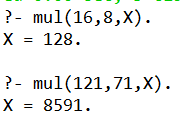
****

**Que 23 WAP to multiply two numbers.**

**Ans**

mul(X,Y,R):- R is X\*Y.

**#Output**

****

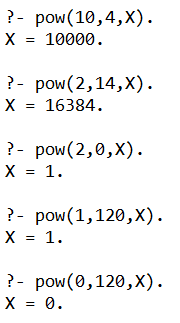
**Que 24 WAP to find the value of first number raised to the power of second number.**

**Ans**

pow(\_,0,R):- R is 1,!.

pow(X,N,R):- N>0,N1 is N-1,pow(X,N1,R1),R is R1\*X.

**#Output**

****

**Que 25 WAP to check if a given string is a palindrome or not**

**Ans**

app([],X,[X]).

app([H|T1],X,[H|T2]):- app(T1,X,T2).

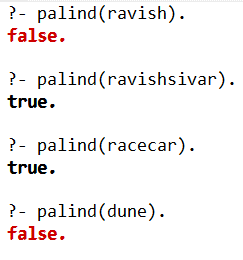
reverse([],[]).

reverse([X],[X]).

reverse([H|T],A):- reverse(T,R1),app(R1,H,A).

palind(R):-atom\_chars(R,R1),reverse(R1,R1),!.

**#Output**

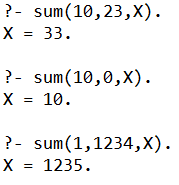


**Que 26 WAP to perform sum of two numbers**

**Ans**

sum(N1,N2,R) :- R is N1+N2.

**#Output**

****

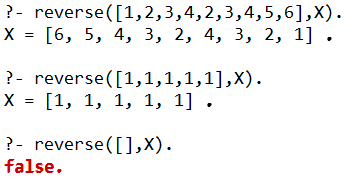
**Que 27 WAP to reverse a list**

**Ans**

reverse([H],[H]).

reverse([H|T],R) :- reverse(T,R1),append(R1,H,R).

**#Output**

****

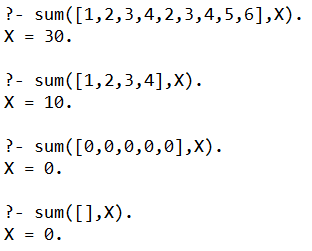
**Que 28 WAP to calculate sum of a list**

**Ans**

sum([],0).

sum([H|T],R):- sum(T,R1), R is H+R1.

**#Output**

****

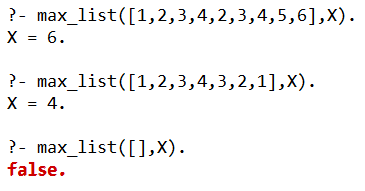
**Que 29 WAP to get max from a list**

**Ans**

max\_list([H],H).

max\_list([H|T],R):- max\_list(T,R1), R1>=H ->R is R1;R is H .

**#Output**

****

**Que 30 WAP 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.**

**Ans**

merge([], Ls, Ls).

merge(Rs, [], Rs).

merge([X1 | Xs1], [X2 | Xs2], [X1 | Ms]) :-

  X1 < X2, !,

  merge(Xs1, [X2 | Xs2], Ms).

merge([X1 | Xs1], [X2 | Xs2], [X2 | Ms]) :-

  X1 >= X2, !,

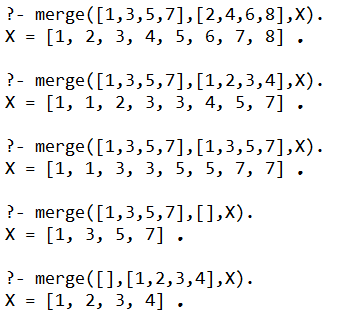
  merge([X1 | Xs1], Xs2, Ms).

merge([X1 | Xs1], [X2 | Xs2], [X1, X2 | Ms]) :-

  X1 = X2, !,

  merge(Xs1, Xs2, Ms).

**#Output**

****

**Que 31 WAP to show family relation.**

**Ans**

parent(sohan,priya).

parent(priya,bobby).

parent(tilak,bobby).

parent(tilak,lalita).

parent(bobby,anita).

parent(bobby,pratiksha).

parent(pratiksha,john).

male(sohan).

male(tilak).

male(bobby).

male(john).

female(priya).

female(lalita).

female(anita).

female(pratiksha).

child(X,Y):- parent(Y,X).

mother(X,Y):- parent(X,Y),female(X).

father(X,Y):- parent(X,Y),male(X).

sister(X,Y):- parent(Z,X),parent(Z,Y),X\==Y,female(X).

brother(X,Y):- parent(Z,X),parent(Z,Y),X\==Y,male(X).

grandparent(X,Y):- parent(X,Z),parent(Z,Y).

grandchild(X,Y):- grandparent(Y,X).

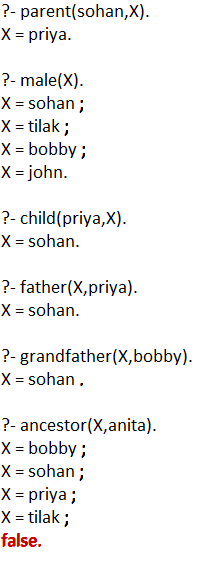
grandfather(X,Y):- grandparent(X,Y),male(X).

grandmother(X,Y):- grandparent(X,Y),female(X).

ancestor(X,Y):- parent(X,Y).

ancestor(X,Y):- parent(X,Z),ancestor(Z,Y).

**#Output**

****