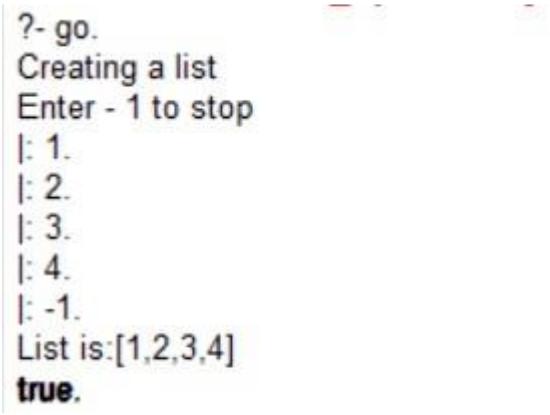
U18CO018

Shubham Shekhaliya

PPL Assignment 4

1. To input the list from the user and print it (Hint: Use read/2 to input the list).

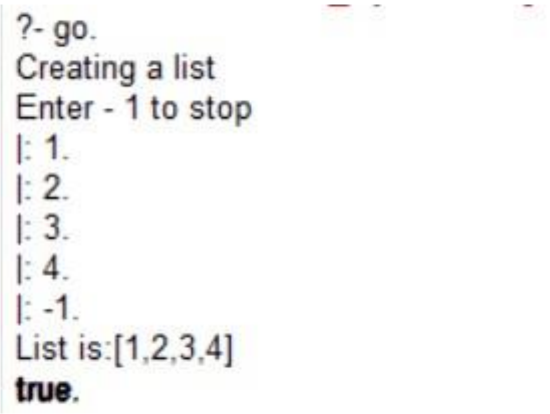
```
create(L1):-
read(Elem),
create(Elem,L1).

create(-1,[]):-!.

create(Elem,[Elem|T]):-read(Next),create(Next,T).

go:- write('Creating a list'),
   nl,
   write('Enter - 1 to stop'),
   nl,
   create(L),
write('List is:'),
write(L).
```



```
?- go.
Creating a list
Enter - 1 to stop
|: 1.
|: 2.
|: 3.
|: 4.
|: -1.
List is:[1,2,3,4]
true.
```

2. Find the sum of all elements in the list.

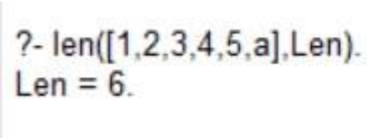
```
sum_list([], 0). sum_list([H|T],
Sum) :-    sum_list(T, Rest),
   Sum is H + Rest.
```



```
?- go.
Creating a list
Enter - 1 to stop
|: 1.
|: 2.
|: 3.
|: 4.
|: -1.
List is:[1,2,3,4]
true.
```

3. Find the size of a list.

```
len([], Ans):-
   Ans is 0.


len([_|Y], Ans):-
   len(Y, L),
   Ans is L + 1.
```

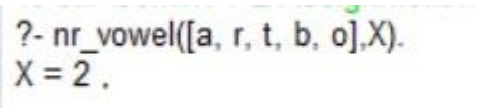


```
?- len([1,2,3,4,5,a],Len).
Len = 6.
```

4. Count no. of vowels in a list.

(Hint: Input list of characters from a user and

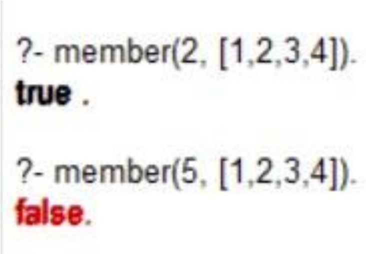
```
count no of vowels in it) vowel(X):-
member(X,[a,e,i,o,u]).

nr_vowel([],0).

nr_vowel([X|T],N):-
   vowel(X),
nr_vowel(T,N1),
   N is N1+1.

nr_vowel([_|T],N):- nr_vowel(T,N).
```



```
?- nr_vowel([a, r, t, b, o],X).
X = 2 .
```

5. Search whether an element exists in a list.

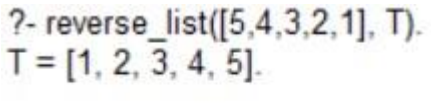
```
member(X,[X|_]).

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



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

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

6. Reverse a given list.

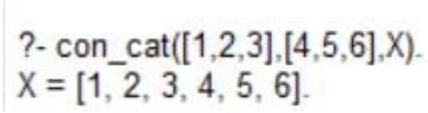
```
reverse_list(Inputlist,Outputlist):-
   reverse(Inputlist,[],Outputlist).

reverse([],Outputlist,Outputlist).
reverse([Head|Tail],List1,List2):-
reverse(Tail,[Head|List1],List2).
```



```
?- reverse_list([5,4,3,2,1], T).
T = [1, 2, 3, 4, 5].
```

7. Concatenate two lists. (Hint: Take two lists namely, L1 and L2 from a user and concatenate it in a list L)

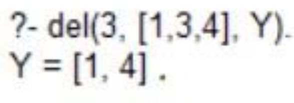
```
con_cat([],L2,L2).

con_cat([H|T],L2,[H|L3]):-
con_cat(T,L2,L3).
```



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

8. Delete an element from the list.

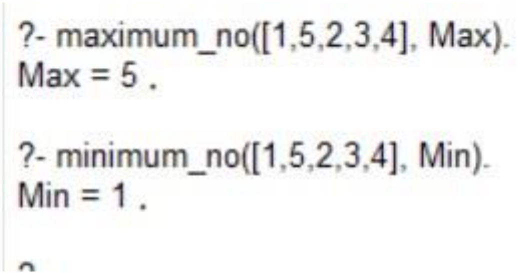
```
del(X,[X|Tail],Tail).

del(X,[Y|Tail],[Y|Tail1]):-
del(X,Tail,Tail1).
```



```
?- del(3, [1,3,4], Y).
Y = [1, 4] .
```

9. Find Max and min elements from the list.

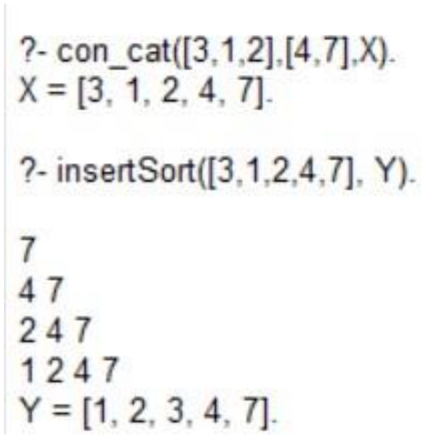
```
maximum_no([X],X).
maximum_no([H|T],Max):-
maximum_no(T,Max),
   H < Max.

maximum_no([Max|T],Max):-
maximum_no(T,M),
   M < Max.

minimum_no([X],X).

minimum_no([H|T],Min):-
minimum_no(T,Min),
   H > Min.

minimum_no([Min|T],Min):-
minimum_no(T,M),
   M > Min.
```



```
?- maximum_no([1,5,2,3,4], Max).
Max = 5 .

?- minimum_no([1,5,2,3,4], Min).
Min = 1 .

?
```

10. Merge and sort two given lists in the third list.

```
con_cat([],L2,L2).
con_cat([H|T],L2,[H|L3]):-
con_cat(T,L2,L3).

insertSort([H|List], Result) :-
insertSort(List, Temp),
printlist(Temp),     insertItem(H,
Temp, Result).

insertSort([], []).

insertItem(X, [H|List], [H|Result]) :-
H < X, !,
   insertItem(X, List, Result).

insertItem(X, List, [X|List]).

printlist([]) :-
   nl.

printlist([X|List]) :- write(X),write(" "), printlist(List).
```

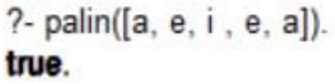


```
?- con_cat([3,1,2],[4,7],X).
X = [3, 1, 2, 4, 7].

?- insertSort([3,1,2,4,7], Y).

7
4 7
2 4 7
1 2 4 7
Y = [1, 2, 3, 4, 7].
```

11. Check if a given list is a palindrome.

```
palin(L):-
```

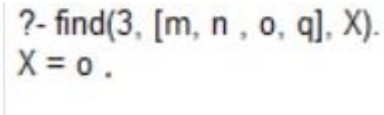
```
reverse(L,L).
```



```
?- palin([a, e, i , e, a]).
true.
```

12. Find an nth element of the list.

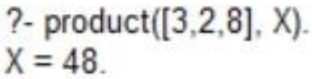
```
find(1,[X|L],X).


find(N,[Y|L],X):-

N1 is N-1,

find(N1,L,X).
```



```
?- find(3, [m, n , o, q], X).
X = o .
```

13. Find the product of all elements in the list.

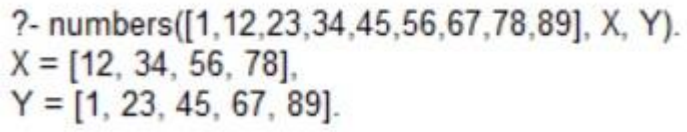
```
product([], 0).


product([H|T], P) :-

product_1(T, H, P). product_1([],

P, P).


product_1([H|T], H0, P) :-

product_1(T, H, P0),

   P is P0 * H0.
```



```
?- product([3,2,8], X).
X = 48.
```

14. Split the list into two parts. Take list L from the user. The list L1 contains all even elements of the list L and the list L2 contains the all odd elements of list L.

```
numbers(L1,L2,L3):-
findall(X,(member(X,L1), X mod 2=:=0),L2),
findall(X,(member(X,L1), X mod 2 =\= 0),L3).
```



```
?- numbers([1,12,23,34,45,56,67,78,89], X, Y).
X = [12, 34, 56, 78],
Y = [1, 23, 45, 67, 89].
```