**Mini Assignment**

**Submitted by: Sarvansh Prasher**

Q.1 Find the last element of a list.

Ans. **Fact & Rules:**

last(Z,[Z]).

last(Z,[\_|T]) :- last(Z,T).

**Sample run queries and results:**

?- last(3,[1,2,3])

**true**

?- last(l,[l,2,3,'Harvard','adam',1.40,l])

**false**

Q.2 Find the last but one element of a list**.**

Ans. **Fact & Rules:**

second\_last(Z,[Z,\_]).

second\_last(Z,[\_,X|Y]) :- second\_last(Z,[X|Y]).

**Sample run queries and results:**

?- second\_last(1,[3,4,2.5,5,3.0,1,9])

**true**

?- second\_last(a,[x,c,2.5,a,3.0,b,9])

**false**

Q.3 **Find the K'th element of a list.**

Ans. **Fact & Rules:**

element\_at(Z,[Z|\_],1).

element\_at(Z,[\_|T],X) :- X > 1 , X1 is X - 1,element\_at(Z,T,X1).

**Sample run queries and results:**

?- element\_at(X,[a,b,c,d,e],1).

**X = a**

?- element\_at(a,[a,b,c,d,e,'Henry'],1).

**true**

Q.4 **Find the number of elements of a list.**

Ans. **Fact & Rules:**

list\_length(0,[]).

list\_length(N,[\_|L]) :- list\_length(N1,L),N is N1 +1.

**Sample run queries and results:**

?- list\_length(X,[a,b,c]).

**X = 3**

?- list\_length(X,['adam','john',a,b,c,1.0]).

**X = 6**

Q.5 **Reverse a list.**

Ans. **Fact & Rules:**

reverse\_list([ ], [ ]).

reverse\_list([H|T], Z) :-reverse\_list(T, Z1), append(Z1, [H], Z).

**Sample run queries and results:**

?- reverse\_list(X,[1,2,3])

**X = [3,2,1]**

?- reverse\_list([d,c,b,a],[a,b,c,d])

**true**

Q.6 **Find out whether a list is a palindrome.**

Ans. **Fact & Rules:**

palindrome([]).

palindrome(L) :- reverse(L,L).

**Sample run queries and results:**

?- palindrome([d,c,b,d])

**false**

?- palindrome([d,b,b,d])

**true**

Q.7 **Flatten a nested list structure.**

Ans. **Fact & Rules:**

flatten([],[]).

flatten([H|T],L) :- flatten(H,Z) , flatten(T,Z1) , append(Z,Z1,L).

flatten(Z,[Z]) :- \+ is\_list(Z).

**Sample run queries and results:**

?- flatten([a, [4, [2, 8], b]], Z)

**Z = [a,4,2,8,b]**

?- flatten([a, ['pan', [2, 'john'], 'adam']], Z).

**Z = [a,pan,2,john,adam]**

Q.8 **Eliminate consecutive duplicates of list elements.**

Ans. **Fact & Rules:**

duplicates([],[]).

duplicates([Z,Z|Z1],L) :- duplicates([Z|Z1],L).

duplicates([Z|Z1], [Z|L]) :- duplicates(Z1, L).

**Sample run queries and results:**

?- duplicates([a,a,a,a,b,c,c,a,a,d,e,e,e,e],Z).

**Z = [a,b,c,a,d,e]**

?- duplicates([1,3,3,a,b,c,c,1,1,d,e,4,e,e],Z).

**Z = [1, 3, a, b, c, 1, d, e, 4, e]**

Q.9 **Pack consecutive duplicates of list elements into sublists.**

Ans. **Fact & Rules:**

sublist([], []).

sublist([H,H|T], Z) :-

sublist([[H,H]|T],Z).

sublist([[H|Hs]|[H|T]], Z) :- sublist([[H,H|Hs]|T], Z).

sublist([H|T], [[H]|Z]) :- sublist(T,Z) ,

not(is\_list(H)).

sublist([H|T], [H|Z]) :- sublist(T,Z).

**Sample run queries and results:**

?- sublist([1,3,4,5,6,3,3,5,5,7,7,9],Z).

**Z = [[1], [3], [4], [5], [6], [3, 3], [5, 5], [7, 7], [9]]**

?- sublist([1,3,3,a,b,c,c,1,1,d,e,4,e,e],Z).

**Z = [[1], [3, 3], [a], [b], [c, c], [1, 1], [d], [e], [4], [e, e]]**

Q.10 **Run-length encoding of a list.**

Ans. **Fact & Rules:**

sublist([], []).

sublist([H,H|T], Z) :-

sublist([[H,H]|T],Z).

sublist([[H|Hs]|[H|T]], Z) :- sublist([[H,H|Hs]|T], Z).

sublist([H|T], [[H]|Z]) :- sublist(T,Z) ,

not(is\_list(H)).

sublist([H|T], [H|Z]) :- sublist(T,Z).

encoding([], []).

encoding(L, [[Z,Z1]|T1]) :- sublist(L,[[Z1|Xs]|T]),length([Z1|Xs],Z),encoding(T, T1)

**Sample run queries and results:**

?- encoding([1,3,4,5,6,3,3,5,5,7,7,9],Z).

**Z = [[1, 1], [1, 3], [1, 4], [1, 5], [1, 6], [2, 3], [2, 5], [2, 7], [1, 9]]**

?- encoding([1,3,3,a,b,c,c,1,1,d,e,4,e,e],Z). **Z = [[1, 1], [2, 3], [1, a], [1, b], [2, c], [2, 1], [1, d], [1, e], [1, 4], [2, e]]**