

- 1) Delete last 3 elements using conc.
 $\Rightarrow \text{delete last 3}(L, 4) :- \text{conc}(L, [-, -, -], L)$
- 2) Delete first and last 3 elements
 $\Rightarrow \text{delete first last 3}([-, -, - | \text{End}], \text{Middle})$
 $\quad :- \text{append}(\text{Middle}, [-, -, -], \text{End})$
- 3) Iteration to add item at last
 $\Rightarrow \text{addend}(x, L, L_1) :- \text{conc}(L, [x], L_1)$
- 4) Remove all items x from a list.
 $\Rightarrow \text{Delete } [-, [], []]$
 $\quad \text{Delete } (x, [x | \text{tail}], \text{tail})$
 $\quad \quad :- \text{delete } [\text{~~x~~ } x, \text{tail}, \text{tail}]$
 $\quad \text{Delete } (x, [y | \text{tail}], [y | \text{tail}])$
 $\quad \quad :- \text{delete } (x, \text{tail}, \text{tail})$
- 5) Reverse a list
 $\rightarrow \text{reverse}([], Y, R) :- R = Y$
 $\text{reverse}([H | T], Y, R) :- \text{reverse}(T, [H | Y], R)$

6) Palindrome predicate:

$\Rightarrow \text{accRev}([], A, A)$

$\text{accRev}([H|T], A, R) :- \text{accRev}(T, [H|A], R)$

$\text{rev}(L, R) :- \text{accRev}(L, [], R)$

$\text{Palindrome}(\text{List}) :- \text{rev}(\text{List}, \text{List})$

7) Maximum of 2 elements

$\Rightarrow \text{max}(X, Y, X) :- X \geq Y$

$\text{max}(X, Y, Y) :- X < Y$

8) Maximum in a list

$\Rightarrow \text{maxList}([X], X)$

$\text{maxList}([X, Y | \text{Rest}], \text{Max})$

$:- \text{maxList}([Y | \text{Rest}], \text{MaxRest}),$
 $\text{max}(X, \text{MaxRest}, \text{Max})$

9) Sum of list

$\Rightarrow \text{sumList}([], 0)$

$\text{sumList}([_ | \text{Tail}], \text{sum})$

$:- \text{sumList}(\text{Tail}, \text{sum}), \text{sum} \text{ is } + \text{sum}$

10) Find if list is ordered.

$\Rightarrow \text{ordered}([X])$

$\text{ordered}([X, Y | \text{Tail}])$

$:- X \leq Y, \text{ordered}([Y | \text{Tail}])$

11) Factorial of number.

⇒ factorial (0, 1)

factorial (N, M) :- N > 0, N is N-1,

factorial (N, M1), M1 is N * M1,

12) Sum of odd, even no.s in a list

is even (N) :- 0 is mod (N, 2)

sum ([], [0, 0])

sum ([H|T], [even, odd])

:- sum (T, [even1, odd1]),

is even (H), even is even1 + H

sum ([H|T], [even, odd])

:- sum (T, [even, odd1]),

odd is odd1 + H

13) Make list into palindrome

reverse ([], Y, R) :- R = Y

reverse ([H|T], Y, R) :- reverse (T, [H|Y], R)

makepalm (X, L) :-

reverse (X, Y, R), conc (X, R, L).