

Prolog Examples

Question 1

- Write a goal to delete the last three elements from a list, L, to produce another list.
- Sample goal: deleteLastThree([1,2,3,4,5,6,7,8,9], L).
- L=[1,2,3,4,5,6]

```
Domains
    lst = Integer*
Predicates
    nondeterm append(lst, lst, lst).
    nondeterm deleteLastThree(lst, lst).
Clauses
    append([], L, L).
    append([H|T], L2, [H|T1]):- append(T, L2, T1).

    deleteLastThree(L, R):- append(R, [_,_,_], L).
Goal
    deleteLastThree([1,2,3,4,5,6,7,8,9], L).
```

Question 2

- Write a goal to delete the first three elements and the last three elements from a list, L, to produce another list, R.
- Sample goal: deleteFirstLastThree([1,2,3,4,5,6,7,8,9], R).
- R=[4,5,6]

```
Domains
    lst = Integer*
Predicates
    nondeterm append(lst, lst, lst).
    nondeterm deleteFirstLastThree(lst, lst).
Clauses
    append([], L, L).
    append([H|T], L2, [H|T1]):- append(T, L2, T1).

    deleteFirstLastThree(L, R):- append([_, _ , _|R], [_, _ , _], L).
Goal
    deleteFirstLastThree([1,2,3,4,5,6,7,8,9], R).
```

Question 3

- Define the predicate last(Item, List) which asserts that Item is the last element of a list, R.
- Write 2 versions

- Using append
- Without using append

```

Domains
    lst = Integer*
Predicates
    nondeterm append(lst, lst, lst).
    nondeterm last(Integer, lst).
Clauses
    append([], L, L).
    append([H|T], L2, [H|T1]):- append(T, L2, T1).

% Using append
last(I, L) :- append(_, [I], L).

% Without using append
%last(I, [I]).
%last(I, [_|T]):- last(I, T).
Goal
    last(5, [1,2,3,4,5]).

```

Question 4

- Define the predicate shift(List1, List2) which asserts that List2 is List1 shifted rotationally by one-element to the left.
- For example:
 - ?- shift([1, 2, 3, 4, 5], L1), shift(L1, L2).
 - L1 = [2, 3, 4, 5, 1]
 - L2 = [3, 4, 5, 1, 2]

```

Domains
    lst = Integer*
Predicates
    nondeterm append(lst, lst, lst).
    nondeterm shift(lst, lst).
Clauses
    append([], L, L).
    append([H|T], L2, [H|T1]):- append(T, L2, T1).

    shift([H|T], L):- append(T, [H], L).
Goal
    shift([1, 2, 3, 4, 5], L1), shift(L1, L2).

```

Question 5

- Define two predicates evenlength(List) and oddlength(List) that assert that their arguments are lists of even or odd length respectively.

```

Domains
    lst = Integer*
Predicates
    nondeterm evenlength(lst).
    nondeterm oddlength(lst).
Clauses
    evenlength([]).
    evenlength([_,_|T]):- evenlength(T).

    oddlength([_]).
    oddlength([_,_|T]):- oddlength(T).
Goal
    %evenlength([1,2,3,4,5,6,7,8,9,10]).
    oddlength([1,2,3,4,5,6,7,8,9,10]).

```

Question 6

- Define the predicate `reverse(List, ReversedList)` that asserts that `ReversedList` is a list whose elements are in the opposite order to `List`.

```

Domains
    lst = Integer*
Predicates
    nondeterm append(lst, lst, lst).
    nondeterm reverse(lst, lst).
Clauses
    append([], L, L).
    append([H|T], L2, [H|T1]):- append(T, L2, T1).

    reverse([], []).
    reverse([H|T], R):- reverse(T, R1), append(R1, [H], R).
Goal
    reverse([1,2,3,4,5,6,7,8,9,10], R).

```

Question 7

Define the predicate `reverse(List, ReversedList)` that asserts that `ReversedList` is a list whose elements are in the opposite order to `List`. Use an accumulator.

```

Domains
    lst = Integer*
Predicates
    nondeterm append(lst, lst, lst).
    nondeterm reverse_aux(lst, lst, lst).
    nondeterm reverse(lst, lst).
Clauses
    append([], L, L).
    append([H|T], L2, [H|T1]):- append(T, L2, T1).

```

```
reverse(L, R):- reverse_aux(L, [], R).
reverse_aux([], R, R).
reverse_aux([H|T], A, R):- reverse_aux(T, [H|A], R).
Goal
reverse([1,2,3,4,5,6,7,8,9,10], R).
```

Question 8

- Define a predicate `sumlist(List, Sum)` that asserts that `List` is a list of numbers and `Sum` is their sum. Use an accumulator.

```
Domains
lst = Integer*
Predicates
nondeterm sumlist_aux(lst, Integer, Integer).
nondeterm sumlist(lst, Integer).
Clauses
sumlist(L, S):- sumlist_aux(L, 0, S).
sumlist_aux([], A, A).
sumlist_aux([H|T], A, S):- NA = A + H, sumlist_aux(T, NA, S).
Goal
sumlist([1,2,3,4,5,6,7,8,9,10], S).
```