**EXPERIMENT – 18**

**Aim:**

Write a prolog program to delete the first occurrence and also all occurrences of a particular

element in a given list.

**Description:**

For deleting single occurrence:

Suppose we have a list L and an element X, we have to delete X from L. So there are three cases −

* If X is the only element, then after deleting it, it will return empty list.
* If X is head of L, the resultant list will be the Tail part.
* If X is present in the Tail part, then delete from there recursively.

For deleting all occurrences:

When the “input” list is empty, then the output list is “empty”.

delete\_all(\_Head, [], []).

When the heads of the “input” and “output” lists don’t match

the element being deleted:

delete\_all(Item, [Head|Tail], [Head|New\_Tail]) :-

Item \= Head, delete\_all(Item, Tail, New\_Tail).

When the heads of the “input” list matches the element being

deleted:

delete\_all(Item, [Item|Tail], New\_Tail) :-

delete\_all(Item, Tail, New\_Tail).

**Program 1:**

(Program to delete first occurrence of an element)

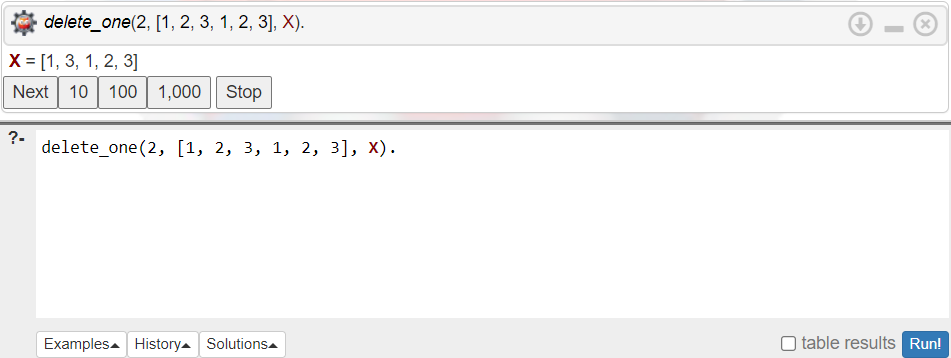
delete\_one(\_, [], []).

delete\_one(Term, [Term|Tail], Tail).

delete\_one(Term, [Head|Tail], [Head|Result]) :-

delete\_one(Term, Tail, Result).

**Output 1:**

****

**Program 2:**

(Program to delete all occurrences of an element)

delete\_all(\_Head, [], []).

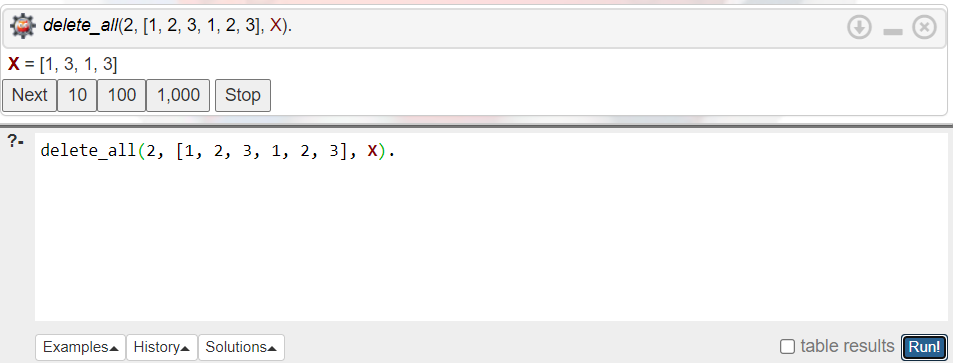
delete\_all(Item, [Head|Tail], [Head|New\_Tail]) :-

Item \= Head, delete\_all(Item, Tail, New\_Tail).

delete\_all(Item, [Item|Tail], New\_Tail) :-

delete\_all(Item, Tail, New\_Tail).

**Output 2:**

****