**Aim:** To implement –

1. Search nth element from a list
2. Add two numbers
3. Find minimum and maximum from a given list

**Theory:**

In this experiment we are performing three different types of programs. One is nth element where we have to search for an element from a list and giving the result. This is not performed using any sorting technique. The element to be searched is done by searching the entire list from position 0 to n-1.

The second type of program is adding up two numbers. This program is as simple as it looks. Like taking two numbers and adding them. But one modification done in this program is taking user input which is done using read and write.

The third type of code is finding the minimum and the maximum of the numbers from the list. This is also not done using any sorting technique. it searches from the list and provides with the result.

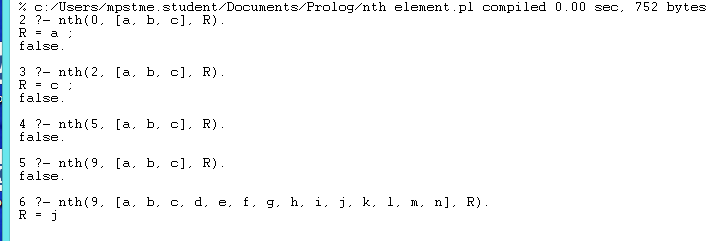
**Code:**

nth element.pl:

nth(0,[X|\_],X).

nth(N,[\_|T],R):- M is N-1, nth(M,T,R).

**Output:**



**Code:**

Add.pl:

start:- sum, nl.

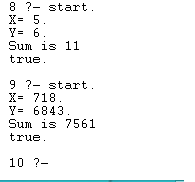
sum:- write('X= '), read(X),

write('Y= '), read(Y),

S is X+Y,

write('Sum is '), write(S).

**Output:**



**Code:**

Minmax.pl

min(A, B, C) :- A < B, !, C = A.

min(\_, B, B).

max(A, B, C) :- A > B, !, C = A.

max(\_, B, B).

minimumAcc(A, [], A) :- !.

minimumAcc(A, [H|T], M) :- min(A, H, B), minimumAcc(B, T, M).

maximumAcc(A, [], A) :- !.

maximumAcc(A, [H|T], M) :- max(A, H, B), maximumAcc(B, T, M).

minimum([H|T], Min) :- minimumAcc(H, T, Min).

maximum([H|T], Max) :- maximumAcc(H, T, Max).

**Output:**

?- consult(minmax).

% minmax compiled 0.00 sec, 11 clauses

true.

?- minimum([5,3,7,6,4,1,9,8,2], X).

X = 1.

?- maximum([5,3,7,6,4,1,9,8,2], X).

X = 9.

**Conclusion:** Hence we have implemented nth element, adding two numbers and finding the minimum and the maximum from a list in prolog and also done it by taking user input.