```prolog
/*
Small set of commands demonstrating how to simulate a
one-dimensional array in Prolog and how to access
individual elements with indexing.

Michael E. Sparks, 6 Nov 2020

SAMPLE USAGE:

Sort elts of list, then take product of 4th & 7th sorted numbers:

?- nth_sorted_elt([6,5,4,4**2,1,12,-5,24],4,Elt1,Sorted),
|    arg(7,Sorted,Elt2),
|    Res is Elt1 * Elt2.
Elt1 = 5,
Sorted = sa(-5, 1, 4, 5, 6, 12, 4**2, 24),
Elt2 = 4**2,
Res = 80.
*/

nth_sorted_elt(List,N,Elt,Sorted) :-
    quicksort(List,List1),
    length(List1,Len),
    functor(Sorted,sa,Len),
    copy_list_to_array(List1,Sorted,1),
    !,
    arg(N,Sorted,Elt).

copy_list_to_array([],_,_).

copy_list_to_array([X|T],A,I) :-
    arg(I,A,X),
    I1 is I + 1,
    copy_list_to_array(T,A,I1).

quicksort([],[]).

quicksort([Pivot|Tail],Sorted) :-
    partition(Pivot,Tail,Smaller,Larger),
    quicksort(Smaller,SortedSmaller),
    quicksort(Larger,SortedLarger),
    append(SortedSmaller,[Pivot|SortedLarger],Sorted).

partition(_,[],[],[]).

partition(Pivot,[X|T],[X|Smaller],Larger) :-
    X =< Pivot, !,
    partition(Pivot,T,Smaller,Larger).

partition(Pivot,[X|T],Smaller,[X|Larger]) :-
    X > Pivot, !, % the .GT. check's technically unnecessary
    partition(Pivot,T,Smaller,Larger).
```