Don’t show your code to anyone in the class. Don’t read anyone else’s code. Do not google to find complete solutions to the exercises (as that involves reading someone else’s code). Don’t discuss the details of your code with anyone except the tutors, grader, or your instructor.

A list is either empty or it is composed of a first element (head) and a tail, which is a list itself. In Prolog we represent the empty list by the atom [] and a non-empty list by a term [H|T] where H denotes the head and T denotes the tail.

1. (2 points) Find the last element of a list.
   Example:
   `?- my_last(X,[a,b,c,d]).`
   X = d

2. (2 points) Find the K'th element of a list.
   The first element in the list is number 1.
   Example:
   `?- element_at(X,[a,b,c,d,e],3).`
   X = c

3. (2 points) Find the number of elements of a list.

4. (3 points) Flatten a nested list structure.
   Transform a list, possibly holding nested lists as elements into a 'flat' list by replacing each list with its elements (recursively).

   Example:
   `?- my_flatten([a, [b, [c, d]], e], X).`
   X = [a, b, c, d, e]

   Hint: Use the predefined predicates is_list/1 and append/3

5. (3 points) Eliminate consecutive duplicates of list elements.
   If a list contains repeated elements they should be replaced with a single copy of the element. The order of the elements should not be changed.

   Example:
   `?- compress([a,a,a,a,b,c,c,a,a,d,e,e,e,e],X).`
   X = [a,b,c,a,d,e]

6. (3 points) Pack consecutive duplicates of list elements into sublists.
If a list contains repeated elements they should be placed in separate sublists.

Example:
?- pack([a,a,a,b,c,c,a,a,d,e,e,e,e],X).
X = [[a,a,a,a],[b],[c,c],[a,a],[d],[e,e,e,e]]

7. (2 points) Duplicate the elements of a list.
Example:
?- double([a,b,c,c,d],X).
X = [a,a,b,b,c,c,c,c,d,d]

8. (3 points) Duplicate the elements of a list a given number of times.
Example:
?- copy([a,b,c],3,X).
X = [a,a,a,b,b,b,c,c,c,c]

What are the results of the goal:
?- copy(X,3,Y).