Name of the Student: _____

Q 1

The built-in function `LAST`, when applied to a list, returns the final element of the list in a list – try and see. Write a recursive function `C-LAST`, which returns the final element – the element itself not a list containing it, if there is one; return `nil`, otherwise.

Q 2

Write a recursive function `C-REMOVE`, custom version of the built-in `REMOVE`, which removes all occurrences of its first argument from the second.

Q 3

Write a recursive function `REM-FIRST`, which removes the first occurrence of its first argument from the second.

Q 4

Write a recursive function `CHOP-END`, which removes the final element of the given list – its like `CDR` from the back. You are NOT allowed to make `(REVERSE (CDR (REVERSE LST)))`. Nothing to be done for an empty list, just return it as it is; but a single element list gets "nilled".

Q 5

Write a recursive function `SUM` that sums the integers in its list argument.

Q 6

Write a recursive function `PALINDROME` that checks whether a given list is a palindrome. You are NOT allowed to use `REVERSE` this time, you need to think recursively. You may need to use some functions you defined above; so put everything in a single file to have everything loaded with a single load command. Here is a hint – you may like not to read it now – (i) an empty list is a palindrome[1]; (ii) a given list is a palindrome, if its first and last elements are the same and what remains in between is a palindrome; (iii) no other list is a palindrome.

Q 7

Write a recursive function `C-LENGTH`, custom version of the built-in `LENGTH`. You will need to keep a counter that increases in every recursion. The counter should start as `0`; therefore make your function 2-place (=getting two arguments as inputs), such that it is always called with `0` as the second argument. What will be your base case? What you will return when you arrive there?

[1] It is, because it is the same whether you read it from left to right or right to left.