Question 1

Write a function `L-PROD`, which takes an integer and a list of integers, and returns a list of integers where each element in the second list got multiplied by the first integer argument.

Question 2

Write a function `PAIR-PROD`, which takes a list of two element lists of integers and returns a list of products of these pairs. E.g. an input like `((7 8) (1 13) (4 1))` should yield `(56 13 4)`.

Question 3

`LISP` has the built-in `NTH`, it takes a position argument and a list and returns the element at the given position. Try and see how it works. Write your own function `NTH2` that behaves exactly like `NTH`, of course without using `NTH`.

Question 4

Write a function `SUBSTITUTE`, with 3 arguments, say `old new exp` such that every occurrence of `old` at the top-level of `exp` is replaced by `new`. By "top-level" we mean the function should not check embedded levels in lists. E.g. `(substitute 'x 'k '(x (x y) z))` should return `(k (x y) z)`.

Question 5

Write a function `LAST-NTH` that returns the *n*th element from the end of a given list. Do NOT use `NTH` or `ELT`; use `DOLIST`.

Question 6

Write a three argument function `REMOVE-N`, which removes every *n*th occurrence of an item from a list. You can use `REM`, which takes 2 number arguments and gives the remainder left when the first is divided by the second.

Question 7

Write a function `MULTI-MEMBER` that checks if its first argument occurs more than once in the second.

Question 8

Write your own function `APPEND2` that appends two list arguments into a third list. You are not allowed to use `APPEND` and `LIST`.

Question 9

Write a function that takes a list of integers and returns the largest integer in the list.

Question 10

Write a function that takes a list of integers and returns the *second* largest integer in the list.