**Problem Sheet 2**

**LISTS**

1. Find the last element of a list.
2. Find the last but one element of a list.
3. Find the K'th element of a list. The first element in the list is number 1.
4. Find the number of elements of a list.
5. Reverse a list.
6. Find out whether a list is a palindrome. A palindrome can be read forward or backward; e.g. (x a m a x).
7. Flatten a nested list structure.

(my-flatten '(a (b (c d) e)))

(A B C D E)

1. Eliminate consecutive duplicates of list elements.

(compress '(a a a a b c c a a d e e e e))

(A B C A D E)

1. Pack consecutive duplicates of list elements into sublists. If a list contains repeated elements they should be placed in separate sublists.

(pack '(a a a a b c c a a d e e e e))

((A A A A) (B) (C C) (A A) (D) (E E E E))

1. Run-length encoding of a list. Use the result of problem P09 to implement the so-called run-length encoding data compression method. Consecutive duplicates of elements are encoded as lists (N E) where N is the number of duplicates of the element E.

(encode '(a a a a b c c a a d e e e e))

((4 A) (1 B) (2 C) (2 A) (1 D)(4 E))