FOLD EXAMPLE

It traverses the list from right to left and applies the combining function.

val fold = fn: ('a -> 'b -> 'b) -> 'b -> 'a list -> 'b

fun f1 x y = x+y;

fun addup L = fold f1 0 L

addup [1,2,3]

1 + (fold f1 0 [2,3])

1 + 2 + (fold f1 0 [3])

1 + 2 + 3 + (fold f1 0 [])

1 + 2 + 3 +

0 1 + 2 + 3

1 + 5

6

MAP

val map = fn: ('a -> 'b) -> 'a list -> 'b list

FILTER

val filter = fn: ('a -> bool) -> 'a list -> 'a list

LAST ELEMENT OF LIST FUN

fun last [] = NONE

| last [x] = SOME(x)

| last (x::rest) = last rest;

val last = fn: 'a list -> 'a option

TREE EXAMPLE

Binary trees with integer data in the interior nodes (and no data at the leaves) can be represented by

datatype tree = LEAF

| NODE of int\*tree\*tree

Give a definition for a function doubleT having type tree -> tree, that makes of copy of the tree given as argument but with all the contained int values doubled.

Fun doubleT (LEAF) = LEAF

| doubleT (NODE(x, t1, t2)) =

NODE(2\*x, doubleT t1, doubleT t2);

