函数式编程原理

实验三

实验目的

- 掌握多态类型、option类型和高阶函数的编程方法

- 利用ML语言求解实际问题

实验内容：

1. 编写函数thenAddOne，要求：
   ① 函数类型为: ((int ->int) * int) -> int；
   ② 功能为将一个整数通过函数变换(如翻倍、求平方或求阶乘)后再加1。

2.（1）编写函数mapList，要求：

① 函数类型为: ((‘a -> ‘b) * ‘a list) -> ‘b list；

② 功能为实现整数集的数学变换(如翻倍、求平方或求阶乘)。

（2）编写函数mapList’，要求：

① 函数类型为: (‘a -> ‘b) -> (‘a list -> ‘b list)；

② 功能为实现整数集的数学变换(如翻倍、求平方或求阶乘)。

③ 比较函数mapList’和mapList，分析、体会它们有什么不同。

3. 编写函数findOdd，要求：

① 函数类型为: int list -> int option；
② 功能为：如果x为L中的第一个奇数，则返回SOME x；否则返回NONE

4. 编写函数：

treeFilter: (‘a -> bool) -> ‘a tree -> ‘a option tree

将树中满足条件P（ ‘a -> bool ）的节点封装成option类型保留，否则替换成NONE。

5. 一棵minheap树定义为：

1. t is Empty;
2. t is a Node(L, x, R), where R, L are minheaps and values(L), value(R) >= x
   (value(T)函数用于获取树T的根节点的值）

编写函数treecompare, SwapDown 和heapify：

treecompare: tree * tree -> order
(* when given two trees, returns a value of type order, based on which tree has a larger value at the root node *)

SwapDown: tree -> tree
(* REQUIRES the subtrees of t are both minheaps
* ENSURES swapDown(t) = if t is Empty or all of t's immediate children are empty then
* just return t, otherwise returns a minheap which contains exactly the elements in t. *)

heapify : tree -> tree
(* given an arbitrary tree *t*, evaluates to a minheap with exactly the elements of *t*. *)

分析SwapDown 和heapify两个函数的work和span。