Your homework is due at 11:59pm on Oct 3, 2018. You will use the Learn OCaml platform to submit and automatically grade your homework. You can submit your homework as often as you like until the due date and immediately check your grade.

Let's have cake!

We will model a cupcake by its price, weight, calories and list of ingredients as per the prelude code. Some examples of cupcakes are also provided.

1. Your task is to implement a function allergy_free : ingredient list -> cupcake list -> cupcake list. It takes a list of ingredients allergens and a list of cupcakes cupcakes as input, and returns the cupcakes from the list that do not contain any of the listed allergens. Cupcakes in the returned list should appear in the same order that they did in the input list. Note that none of the ingredient lists are sorted.

2. allergy_free [Nuts; Gluten] cupcakes;;

   - : cupcake list = [Cupcake (2.75, 90.5, 275, [Dairy; Soy])]

   In order to get full marks, you **must** use each of the following higher-order functions:

- List.filter : ('a -> bool) -> 'a list -> 'a list
- List.exists : ('a -> bool) -> 'a list -> bool
- List.for_all : ('a -> bool) -> 'a list -> bool

Generic Tree Traversals

1.

a. Implement a function tree_map which when given a function f : 'a -> 'b and a tree t : 'a tree applies f to all the entries in the tree.
b. Using tree_map, implement the function delete_data that takes a tree whose nodes are key-value pairs and returns the tree with only the keys.

2.

a. Intuitively, fold_right f e replaces every :: by f and nil by e in a list. The function fold_treefor binary trees is analogous to fold_right. Given a tree, fold_tree f e replaces each Leaf by some value e and each Node by the application of a 3-argument function f

   It has type:

   tree_fold: ('a * 'b * 'b -> 'b) -> 'b -> 'a tree -> 'b

Example: Given a tree

```
Node (x0, Node (x1, Empty
                        , Empty)
              , Node (x2, Node (x3, Empty
                                    , Empty)
                        , Empty))
```

the result will be

```
f(x0, f (x1, e
              , e)
       , f (x2, f (x3, e
                        , e)
              , e))
```

b. The fold_tree function allows us to express many programs which traverse trees elegantly in one line. Using fold_tree:

i. implement the function size : 'a tree -> int which given a binary tree returns the number of nodes in the tree.

ii. implement the function reflect : 'a tree -> int which given a binary tree swaps the left and the right child.

iii. implement inorder: 'a tree -> 'a list which given a binary tree returns a list of all entries in order.