CS2203 / CSX3004 Programming Languages

Week 5 Worksheet

Write the following functions in ML.

1. Function `squarelist` of type `'int list -> 'int list` that takes a list of integers and returns the list of the squares of those integers. For instance, if the input is `[1, 2, 3, 4]`, your function should return `[1, 4, 9, 16]`.

2. Function `multpairs` of type `(int * int) list -> int list` that takes a list of pairs of integers and returns a list of the product of each pair. For instance, if the input is `[(1,2), (3,4)]`, your function should return `[2, 12]`.

3. Function `ilist2rlist` of type `int list -> real list` that takes a list of integers and returns a list of the same numbers converted to type real. For example, if the input is `[1, 2, 3]`, you should get `[1.0, 2.0, 3.0]`.

4. Function `truecount` of type `bool list -> int` that takes a list of Boolean values and returns the number of `true`s in the list.

5. Function `evens` of type `int list -> int list` that converts a list of integers and returns the list of all the event elements from the original list. For instance, if the input is `[1, 2, 3, 4, 5, 6, 7]`, your function should return `[2, 4, 6]`.

6. Function `maxpairs` of type `(int * int ) list -> int list` that takes a list of pairs of integers and returns the list of the max elements from each pair. For example, if the input is `[(1,3), (4,2), (~3,~4)]`, your function should return `[3, 4, ~3]`.

7. Function `convert` of type `('a * 'b) list -> 'a list * 'b list` that converts a list of pairs into a pair of lists, preserving the order of the elements. For example, if the input is `[(1,2), (3,4), (5,6)]`, your function should return `([1, 3, 5], [2, 4, 6])`.

8. Define a function `mymap` with the same type and behavior as `map`, but without using `map`. Note `foldl` or `foldr` is allowed.

9. Define a function `mymap2` with the same type and behavior as `map`, but without using `map`, `foldl` and `foldr`.

10. Define a function `myfoldl` with the same type and behavior as `foldl` but without using `map`, `foldl` and `foldr`.