

COSC 251 – Chapter 5: A First Look At ML

Submit your solution through Blackboard

Assignment

Throughout this chapter, we have used the SML/NJ language system in an interactive mode. For longer examples, it makes more sense to store your function definitions in a file [which you will submit]. Once you have created a file containing a definition or definitions, you can load it into an ML session by using the predefined `use` function. For example, if you have created a file called `assign1.sml` in the current directory, you can run your ML language system and type `use "assign1.sml";` after the prompt. The ML language system will read the contents of the file just as if you had typed it one line at a time. After `use` finishes, you can continue typing interactive ML expressions, for example, to test the function definitions in your file.

Exercise 1 Write a function `cube` of type `int -> int` that returns the cube of its parameter.

Exercise 2 Write a function `cuber` of type `real -> real` that returns the cube of its parameter.

Exercise 4 Write a function `min3` of type `int * int * int -> int` that returns the smallest of three integers.

Exercise 7 Write a function `cycle1` of type `'a list -> 'a list` whose output list is the same as the input list, but with the first element of the list moved to the end. For example, `cycle1 [1,2,3,4]` should return `[2,3,4,1]`.

Exercise 8 Write a function `sort3` of type `real * real * real -> real list` that returns a list of three numbers, in sorted order with the smallest first.

Exercise 9 Write a function `del3` of type `'a list -> 'a list` whose output is the same as the input list, but with the third element deleted. Your function need not behave well on lists with lengths less than 3.

Exercise 10 Write a function `sqsum` of type `int -> int` that takes a non-negative integer n and returns the sum of the squares of all the integers 0 through n . Your function need not behave well on inputs less than zero.

Exercise 11 Write a function `'a list * int -> 'a list` that takes a list and an integer n as input and returns the same list, but with the first element cycled to the end of the list n times. (Make use of your `cycle1` function from a previous exercise.) For example, `cycle([1,2,3,4,5,6],2)` should return the list `[3,4,5,6,1,2]`.

Exercise 13 Write a function `max` of type `int list -> int` that returns the largest element of a list of integers. Your function need not behave well if the list is empty. *Hint:* Write a helper function `maxhelper` that takes as a second parameter the largest element seen so far. Then you can complete the exercise by defining

```
fun max x = maxhelper (tl x, hd x);
```

Exercise 14 Write a function `isPrime` of type `int -> bool` that returns true if and only if its integer parameter is a prime number. Your function need not behave well if the parameter is negative.