CS 2041: Practice Exam 2
Fall 2018
University of Minnesota
Exam period: 20 minutes
Points available: 40

Problem 1 (10 pts): Write a function called `strlen_between` which accepts a min and max length and a list of strings. A list of strings with length in the given range is returned. Min/max lengths are inclusive. For full credit, make use of a higher-order function in your definition. Demo uses are given below.

1 # #use "strlen_between.ml";;
2 val strlen_between :
3 int -> int -> string list -> string list = <fun>
4 # let lst = ["aaa";"bbbb"; "cccc";"dddddd"];;
5 val lst : string list = ["aaa"; "bbbb"; "cccc"; "dddddd"]
6 # strlen_between 2 4 lst;;
7 - : string list = ["aaa"; "cccc"]
8 # strlen_between 4 7 lst;;
9 - : string list = ["bbbb"; "cccc"; "dddddd"]
10 # strlen_between 9 12 lst;;
11 - : string list = []

Problem 2 (10 pts): Write a function called `largest_even` which accepts a list of integers. If no even numbers are in the list, return None. Otherwise, return Some of the largest integer in the list. For full credit, use pattern matching and a higher-order function in your solution. Demo uses are given below.

1 # #use "largest_even.ml";;
2 val largest_even :
3 int list -> int option = <fun>
4 # largest_even [];;
5 - : int option = None
6 # largest_even [1;3;5];;
7 - : int option = None
8 # largest_even [4];;
9 - : int option = Some 4
10 # largest_even [1;3;2;5];;
11 - : int option = Some 2
12 # largest_even [1;4;3;2;5;8;7;6];;
13 - : int option = Some 8

Write your code for `strlen_between` here.

Write your code for `largest_even` here.
**Background:** Shae Lowcopy decided to extend the *multimanager* application to allow the current list to be copied using the *copyto* command. She modifies *multimanager.ml* to include the following additional command sequence.

```ml
let execute_command tokens =
  let cmd = tokens.(0) in (* 0th element is command *)
  match cmd with
  | "copyto" ->
    let new_name = tokens.(1) in
    let added = Doccol.add global new_name global.curdoc in
    if added then
      printf "Copied list to '%s'\n" new_name
    else
      printf "ERROR: list '%s' already exists, cannot create copy\n" new_name
  ...
```

Unfortunately when she begins testing she sees the below undesirable behavior: the copied list seems to affect the original list when they should be independent.

**Problem 3 (10 pts):** Describe in some detail why Shae’s implementation of *copyto* does not work as expected and why the two lists seem to be linked somehow.

**Problem 4 (10 pts):** Describe how to fix the problem so that a proper list copy is made for *copyto*. Provide at least some code to give a concrete idea of how your idea would work.