

CS 2041: Practice Exam 1

Fall 2018

University of Minnesota

Exam period: 20 minutes

Points available: 40

Problem 1 (10 pts): Write a function called `even_indices` which takes any type of list and returns a list of elements at even indices 0,2,4,etc. Example uses from a REPL are shown. *Hint: a recursive solution which “skips” element is effective. My if/else solution is 13 lines long while pattern matching makes this considerably shorter.*

Write your code for even_indices here.

```

1 # #use "even_indices.ml";;
2 val even_indices : 'a list -> 'a list = <fun>
3 # even_indices [];;
4 - : 'a list = []
5 # even_indices [0];;
6 - : int list = [0]
7 # even_indices [0; 1];;
8 - : int list = [0]
9 # even_indices [0; 1; 2; 3; 4; 5];;
10 - : int list = [0; 2; 4]
11 # even_indices [0; 1; 2; 3; 4; 5; 6; 7; 8];;
12 - : int list = [0; 2; 4; 6; 8]
13 # even_indices ["a"; "b"; "c"; "d"];;
14 - : string list = ["a"; "c"]

```

Problem 2 (10 pts): Source code for the `array_fill` function is provided along with a short session which attempts to demonstrate the function. A warning is given on loading the code and an unexpected result occurs. Describe the following.

(A) Why is the warning given?

```

> cat -n fill.ml
1 (* fill array with given element *)
2 let fill_array arr elem =
3   for i=0 to (Array.length arr)-1 do
4     arr.(i) = elem;
5   done;
6 ;;

```

(B) Why is the array apparently unchanged?

```

> ocaml
# #use "fill.ml";;
File "fill.ml", line 3, characters 4-18:
Warning 10: this expression should have type unit.
val fill_array : 'a array -> 'a -> unit = <fun>

```

(C) How can the function be corrected to remove the warning and carry out its intended purpose?

```

# let a = [|9;5;2|];;
val a : int array = [|9; 5; 2|]

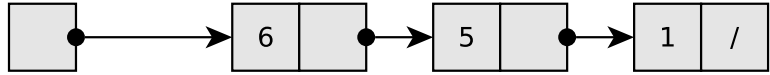
# fill_array a 7;;
- : unit = ()

# a;;
- : int array = [|9; 5; 2|]

```

Problem 3 (10 pts): Complete the pointer diagram to show to reflect how the OCaml code will use existing cons boxes and create new ones.

let listX = [6; 1; 2];;



let ansA = List.hd (List.tl listX);;



let ansB = List.tl (List.tl listX);;



let ansC = 7 :: listX;



let ansD = 9 :: 5 :: ansB;;



Problem 4 (10 pts): Write a function called `firstlast` which returns a list of the first and last elements of a parameter list. For empty lists, the empty list is returned. For single element lists, only that element is returned. For full credit, make use of a **tail-recursive helper function** to complete the function.

Write your code for firstlast here.

```

1 (* REPL demo for firstlast *)
2 # firstlast [];;
3 - : 'a list = []
4 # firstlast ["a"];;
5 - : string list = ["a"]
6 # firstlast ["a";"b"];;
7 - : string list = ["a"; "b"]
8 # firstlast ["a";"b";"c";"d"];;
9 - : string list = ["a"; "d"]
10 # firstlast ["a";"b";"c";"d";"e";"f"];;
11 - : string list = ["a"; "f"]
12 # firstlast [1;2;3;4;5;6];;
13 - : int list = [1; 6]
  
```