Assignment 2

Due: Thursday, February 19, 11:59pm

Some rules and conditions:

1. This assignment is to be done in singles.
2. Submit your work electronically - see instructions for each question.
3. Late submission penalty: 25% off the grade for every 24 hours or part of thereof.
4. Cheating will not be tolerated. Remember, its *very* easy to see when an assignment has been copied.
5. If anything is unclear, email me. Frequently asked questions will be posted on the website.

Question 1 (20 points)**.** Exercise 3.8 from Hickey.

What to submit: an OCaml source file named `stream.ml` which contains the four requested functions `(+:), (-|), map, integral`, and any other functions you use (like `tl` or `hd`). Above each function you define, please include a block of comments that justifies your definition (i.e. explains *why* you defined it in such a way).

How to submit: `submit 3401 a2 stream.ml`

Hints and notes: this question is very similar in spirit to 3.6. You do not need recursion in part 1.

Question 2 (10 points)**.** Exercise 5.7 from Hickey. Note that the definition of `append` provided in the question uses pattern matching, here's an equivalent definition using `if ... then`:

```
let rec append l1 l2 =
  if l1 = [] then l2
  else (List.hd l1)::(append (List.tl l1) l2);;
```

I do not care whether you use pattern matching or not in your definition.

What to submit: an OCaml source file named `append.ml` which contains the requested tail-recursive function `append`, and any other functions you need. You are not allowed to use any `List.` functions, except `List.hd, List.tl`. Briefly explain how each function works inside a block of comments above it.

How to submit: `submit 3401 a2 append.ml`

Hints and notes: to make yourself feel good about what you wrote, test your function on two huge lists. For example, a list with 1000000 zero's can be created like this:

```
let big_list = Array.to_list (Array.make 1000000 0);;
```

So, if your function is indeed tail-recursive, you shouldn't get stack overflows when you run it like this:

```
append big_list big_list;;
```