```
(*
                            CS 51 Lab 15
                         Native Lazy Streams
 *)

(*.....................................................................
An implementation of lazy streams using OCaml's native `Lazy` module,
along with with some useful functions.

See the corresponding .mli file for documentation.
 *)

type 'a stream_internal = Cons of 'a * 'a stream
 and 'a stream = 'a stream_internal Lazy.t ;;

let head (s : 'a stream) : 'a =
  let Cons (hd, _tl) = Lazy.force s in hd ;;

let tail (s : 'a stream) : 'a stream =
  let Cons (_hd, tl) = Lazy.force s in tl ;;

let rec first (n : int) (s : 'a stream) : 'a list =
  if n = 0 then []
  else head s :: first (n - 1) (tail s) ;;

let rec smap (f : 'a -> 'b)
             (s : 'a stream)
           : 'b stream =
  failwith "smap native not implemented" ;;

let rec smap2 (f : 'a -> 'b -> 'c)
              (s1 : 'a stream)
              (s2 : 'b stream)
            : 'c stream =
  failwith "smap2 native not implemented" ;;

let rec sfilter (pred : 'a -> bool) (s : 'a stream) : 'a stream =
  failwith "sfilter native not implemented" ;;
```