

```
(*
                                CS51 Lab 20
                                Synthesis
*)

type image = float list list ;;
(* images are lists of lists of floats between 0. (white) and 1. (black) *)
type size = int * int ;;
open Graphics ;;

(* threshold threshold image -- image where pixels above the threshold
value are black *)
let threshold img threshold =
  List.map (fun row -> List.map (fun v -> if v <= threshold then 0. else 1.)
        row) img

(* show the image *)
let depict img =
  Graphics.open_graph ""; Graphics.clear_graph ();
  let x, y = List.length (List.hd img), List.length img in Graphics.resize_window x y;
  let depict_pix v r c = let lvl = int_of_float (255. *. (1. -. v)) in Graphics.set_col
or (Graphics.rgb lvl lvl lvl);
  plot c (y - r) in
  List.iteri (fun r row -> List.iteri (fun c pix -> depict_pix pix r c) row) img;
  Unix.sleep 2; Graphics.close_graph () ;;

(* dither max image -- dithered image *)
let dither img =
  List.map
    (fun row ->
      List.map
        (fun v -> if v > Random.float 1.
          then 1.
          else 0.) row)
    img

let mona = Monalisa.image ;;

depict mona ;;

let mona_threshold = threshold mona 0.75 ;;
depict mona_threshold ;;

let mona_dither = dither mona ;;
depict mona_dither ;;
```