

(*

CS51 Lab 7
Modules & Functors

A module for colors and color names

The representation for colors in this implementation of the signature in 'color.mli' is really obscure (and arguably unnecessarily so). By the way, it also has some bugs so it doesn't pass all the unit tests. No need to debug it though, or even read it. You'll be replacing it wholesale with your own hopefully simpler implementation. But you'll *definitely* want to look over the 'color.mli' signature. *)

```
type color = int ;;
```

```
type color_name =
```

```
  Red  
  Green  
  Blue  
  Orange  
  Yellow  
  Indigo  
  Violet ;;
```

```
(* to_color r g b -- Returns the 'color' corresponding to the RGB values  
   given by 'r', 'g', and 'b' *)
```

```
let to_color (r : int) (g : int) (b : int) : color =  
  r lsl 0b10000 + g lsl 0b1000 + b ;;
```

```
(* red c / green c / blue c -- Returns the corresponding channel value  
   for the color 'c' *)
```

```
let red_channel (c : color) : int =  
  c lsr 0b10000 ;;
```

```
let green_channel (c : color) : int =  
  (c lsr 0b1000) land 0b11111111 ;;
```

```
let blue_channel (c : color) : int =  
  c land 0b11111111 ;;
```

```
(* color_named name -- Returns the color corresponding to the color  
   'name' *)
```

```
let color_named (name : color_name) : color =
```

```
  match name with  
  | Red -> 0b11111111100000000000000000  
  | Green -> 0b11111111100000000  
  | Blue -> 0b11111111  
  | Orange -> 0b111111111101011010000000  
  | Yellow -> 0b111111111111111110000000  
  | Indigo -> 0b010110110000000010000010  
  | Violet -> 0b111100001000001011110100 ;;
```