CSC104 tutorial exercises # 4

This tutorial will give you a chance to focus on how you come up with an algorithm, and how to manipulate all the colours in an image with a single command.

My office hour: Wednesdays 5:10–6, in SF1101 (our classroom).

Your tutorial: Wednesdays 6:10–7:00. Tutorial sections are as follows:

Surname	Tutorial section	Room	TA
A–C	section 1	BA3175	Omar
D–J	section 2	BA3185	Nahla
K–L	section 3	BA3195	Dhinakaran
M–T	section 4	BA2220	Nick
U–Z	section 5	BA2220	Yashuai

DCS Help Centre: Monday–Thursday, 4–6 pm in BA2230, see Help Centre page. Khaled, a TA from our course, is in the Centre Monday, Tuesday, and Thursday.

1. Download the folding handout and work through at least a few small cases. Although I don't require you to completely solve the problem by Wednesday, you should be able to say what the crease pattern is for 2, 3, and 4 folds.

2. Download the racket code from the October 10th lecture (right-click on the link for the file), and experiment with some of the examples using `map` and `map-image`. Create a list of strings, then use `map` to create the corresponding list of the lengths of those strings.

3. Imitate the function `swap-red-blue` to create `rotate-red-blue-green`:

```
(define c (make-color 1 2 3 4)) ; for testing purposes

; rotate-red-blue-green : color -> color
; Produce new color with red intensity equal to col's
; green intensity, green intensity equal to col's blue
; intensity, and blue intensity equal to col's red intensity.
(check-expect (rotate-red-blue-green c)
              (make-color
                (color-green c)
                (color-blue c)
                (color-red c)
                (color-alpha c)))
;
(define (rotate-red-blue-green col)
   ; this is the part you complete
```

   Now, try out `(map-image rotate-red-blue-green ...)` on some image of your choice.

4. You can find some information on `color` structs in Picturing Programs, Section 20.7. Please note that the author uses a contract for `map-image` that's different from what we're using: he includes the x and y coordinates. Either contract works.