National University of Singapore
School of Continuing and Lifelong Education

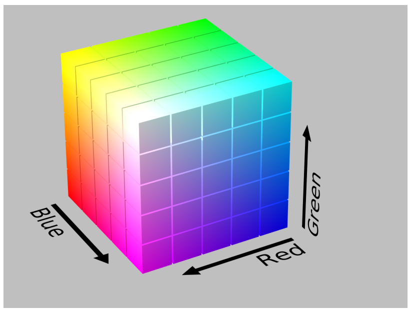
TIC1001: Introduction to Computing and Programming I
Semester I, 2019/2020

Problem Set 2
Branching and Looping

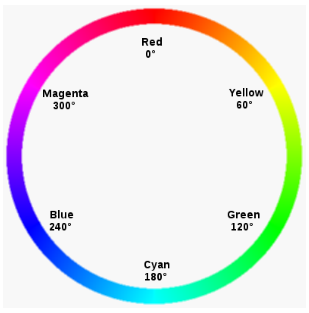
Release date: 22nd August 2019, 8:00 pm
Due: 15th September 2019, 6:00 pm

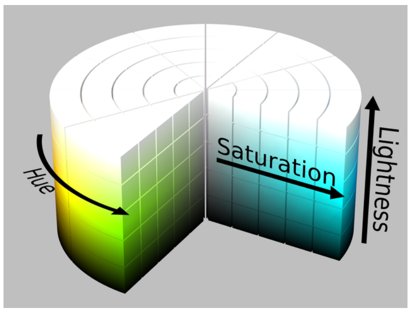
Task 1: Colourspace Conversion (5 marks)

Most of us are familiar with the RGB colour model, where a colour is described by the intensity of red, green and blue colours added together. You can view it as a 3D cube where red, green and blue are on the three different axis.



HSL is another model to describe colour. where the components are Hue, Saturation and Lightness or Luminosity. Hue describes the colour as an angle on a circle, with $0°$ being red, passing through green at $120°$, and blue at $240°$, and then finally back to red. Saturation describes how deep and Lightness describes how bright the colour is.





The intensity for each of the red, green and blue components are usually specified on a 0–255 scale, with 0 being the least and 255 begin 100%. It can be hard to imagine what a colour is based on its RGB value. For example take the colour R=24, G=98, B=118 or in percentage scale, R=9%, G=38%, B=46%. This is a deep cyan colour.

Under the HSL model, the same colour is described as H=$193°$, S=67%, L=28%. The values tell that it is quite saturated and rather dark. The colour wheel shows that $193°$ is pretty close to cyan. So we can guess it is a dark, deep cyan colour.

Converting from RGB to HSL [1]

For this task, you will convert an RGB colour to its HSL equivalent.

1. Scale the RGB values to the range 0–1. Since the inputs are on a 0–255 scale, we can simply divide by 255.
2. Find the maximum and minimum value amongst the R, G and B component.
3. The Luminance value is the midpoint between the max and min values. This can be computed by adding the max and min values and dividing by 2.
4. The next step is to find the Saturation. If the min and max values are equal, this means all three RGB values are equal, so there is no saturation. Everything is a shade of grey. If there is no Saturation, then there is no Hue as well. In this case, Hue is set to $0°$ and saturation to 0.
5. If there is Saturation, the value depends on the level of Luminance:

$$\text{Saturation} = \begin{cases} \dfrac{max - min}{max + min} & \text{if Luminance} < 0.5 \\[2ex] \dfrac{max - min}{2 - max - min} & \text{if Luminance} \geq 0.5 \end{cases}$$

6. Finally to compute Hue, it depends one which of the RGB component is the maximum value:

$$\text{Hue} = \begin{cases} \dfrac{G - B}{max - min} & \text{if Red is max} \\ 2 + \dfrac{B - R}{max - min} & \text{if Green is max} \\ 4 + \dfrac{R - G}{max - min} & \text{if Blue is max} \end{cases}$$

   Multiply the Hue value by 60 to convert it to degrees. If the result is negative, you have to add 360 to adjust it into the correct range.

Implementation

You are to implement the function `rgb_to_hsl`, that takes in 3 integers representing Red, Green and Blue each on a scale of 0–255. The function should convert the RGB colour into its closest HSL representation where $0 \leq$ Hue < 360, and both Saturation and Luminance is on a scale of 0–100. All three values should be rounded to the nearest integer. You can use the `round` function in the `<math.h>` library.

In order to facilitate things, we created a special type for the function to return along with certain code set up in the template. You do not have to be concerned with this. Simply assign the correct values to the int variables `hue`, `sat` and `lum`.

[1] Adapted from http://www.niwa.nu/2013/05/math-behind-colorspace-conversions-rgb-hsl/

Task 2: Taxi Fare (5 marks)

The taxi fare structure in Singapore must be one of the most complex in the world! Check out http://www.taxisingapore.com/taxi-fare/

For the purpose of this exercise, we will just use the following simplified fare structure:

Basic Fare

The first 1 km or less (Flag Down)	$3.40
Every 400 m thereafter or less up to 10 km	$0.22
Every 350 m thereafter or less after 10 km	$0.22

Surcharge

Monday to Friday	6:00 am - 8:59 am	25% of metered fare
Daily	6:00 pm - 11:59 pm	25% of metered fare
Daily	0:00 mn - 5:59 am	50% of metered fare

Note that the surcharge is computed based on the current time. For instance, if the trip started at 5:50 pm and ended at 6:10 pm, only the last 10 mins of the trip will incur the extra 25% surcharge. The surcharge is levied based on the current time when the meter charges for the next distance-block.

We will assume that the distance travelled is proportional to the duration of the trip, i.e. the taxi travels at a constant speed. The speed is given as the number of integer seconds taken to cover 50 m. This ensures that the time taken to travel over a distance-block will be an integer in seconds.

Examine the following examples for more details:

Example 1

Start: Mon 17:59 Speed: 50 m per 6 s (500 m/min) Distance: 1,000 m

17:59:00	$3.40 is added for the next 1 km.
18:01:00	Trip ends with 1 km exactly travelled.

So the total fare is $3.40, even though it ended during the daily even peak surcharge. No surcharge was incurred because the $3.40 was added to the fare before 18:00.

Example 2

Start: Mon 17:57 Speed: 50 m per 6 s (500 m/min) Distance: 2,000 m

17:57:00	$3.40 is added for the next 1 km.
17:59:00	1 km has passed but we have more to go. $0.22 is added for the next 400 m.
17:59:48	1,400 m has passed. $0.22 is added for the next 400 m.
18:00:36	1,800 m has passed. Peak period surcharge started at 18:00:00. $0.275 is added for the next 400 m due to extra 25% surcharge.
18:01:00	Trip ends with 2,000 m travelled. Total fare is $4.115.

Example 3

Start: Mon 05:50   Speed: 50 m per 6 s (500 m/min)   Distance: 15,000 m

05:50:00	$5.10 is added for the first 1 km, due to 50% surcharge.
05:52:00	1 km has passed. $0.33 is added for the next 400 m due to 50% surcharge.
⋮	$0.33 is added for every 400 m travelled.
06:00:00	5,000 m has passed. Surcharge changes to 25%. $0.275 is added for the next 400 m.
⋮	$0.275 is added for every 400 m travelled.
06:09:36	9,800 m has passed. $0.275 is added for next 400 m travelled.
06:10:24	**10,200 m** has passed. $0.275 is added for next **350 m** travelled.
⋮	$0.275 is added for every **350 m** travelled.
06:19:30	14,750 m has passed. $0.275 is added for the last 350 m block.
06:20:00	Trip ends with 15,000 m travelled. Total fare is $15.825.

Implementation

Write a function `double taxi_fare` which takes the following inputs:

- `int weekday` denoting the day of the week, 1–7 being Monday and 7 to Sunday.
- `int start_time` The starting time of the trip, in **number of minutes since midnight** of the stated day.
- `int speed` Given as the time the taxi takes to cover 50 m. i.e., 50 m per x s, where x is the input value.
- `int distance` The distance of the trip, in metres.

The function should return the cost of the fare in dollars.

Hints

1. Although it is possible to compute the fare purely using arithmetic, it is very difficult to do so. It is much easier to “simulate” the actual passage of time and distance of the trip and compute the fare like what happens in real-life.
2. We are only interested in the return value of your function. Since `printf` statement is just a side-effect, you do not need to remove them for submission if you have used them for debugging.
3. This task can seem quite daunting to accomplish in one go. However, it can be reduced to a smaller, more achievable task upon which you can slowly add in the features. For example, you can start out ignoring any surcharge and simply compute the fare based solely on the distance travelled. Once that is done, you can begin to introduce the surcharges, and finally handle all the corner cases.
4. Is your program robust enough to handle unconventional yet valid inputs? While you do not have to handle numerical overflow, it does take a rather large number for that to happen.

Partial marks will be awarded for partially working code. For example, if your code does not handle surcharge and thus only returns correct results for trips between 9 am and 6 pm daily, you will still get partial credit.