

Questão 3 (4,0)

Nessa questão final iremos trabalhar com imagens em cores. Nosso objetivo será criar 4 processamentos de imagens a serem realizados na GPU. Para isto você deverá usar as funções definidas nos arquivos *imagem.cpp/h*.

Processamento 1 - conversão para níveis de cinza

Crie um programa `rgb2gray` que converta uma imagem colorida para níveis de cinza.

```
$> rgb2gray img-in.ppm img-out.pgm
```

A conversão será feita seguindo a seguinte fórmula:

$$G = 0.3R + 0.59G + 0.11B$$

Processamento 2 - filtro de cores

Crie um programa `similar-color` que cria uma imagem de máscara (preta ou branca, somente) contendo os pontos da imagem original que estão próximos o suficiente de uma cor passada pelo usuário. Veja o exemplo de uso abaixo.

```
$> echo "255 255 255" | ./similar-color img-in.ppm img-out.pgm
```

Definimos um pixel p como `255` na imagem de saída se a cor (r_p, g_p, b_p) tiver distância euclidiana menor que 30 em relação à cor referência (r_u, g_u, b_u) .

$$\sqrt{(r_p - r_u)^2 + (g_p - g_u)^2 + (b_p - b_u)^2} < 30$$

Processamento 3 - máscara

Crie um programa `mask` que aplica uma máscara como a criada no item anterior a uma imagem colorida.

```
$> mask img-in.ppm mask.pgm img-out.ppm
```

Cada ponto preto na máscara produz um ponto preto na imagem de saída. Cada ponto branco na máscara produz uma cópia do pixel original na imagem de saída.

Processamento 4 - diferença entre imagens

Crie um programa que faz a subtração entre 2 imagens e guarda em uma terceira imagem. Seu programa deverá aceitar tanto pares de imagens em níveis de cinza e em cores (mas não vale misturar os tipos). A imagem de saída deverá ter o mesmo tipo da imagem de entrada.

```
$> img-diff img-in.ppm img-in2.ppm img-out.ppm
```

Seu programa deverá implementar saturação corretamente. Ou seja, se uma diferença gerar um valor menor que 0 ela deverá ser 0.