

Faça o download do arquivo **imagens.pyc** e das imagens (\*.jpg) para dentro da “Pasta pessoal”.

### Introdução

Antes das máquinas digitais, as fotos eram feitas com máquinas analógicas e filmes fotossensíveis. Um problema que ocorria eventualmente era a sobreposição de fotos. Isso acontecia quando um quadro do filme não era "puxado" corretamente dentro da máquina. Daí, uma foto podia ser "queimada" por cima de outra, resultando em algo como essa foto ao lado.



Nesta prática, você fará um programa para simular essa *dupla exposição fotográfica*. Mas, em vez de lidar com imagens coloridas, faremos isso com duas imagens de mesmo tamanho em tons de cinza. Isso significa que cada *pixel* é formado por um único valor entre 0 (preto) e 255 (branco), em vez dos três componentes R, G, B que usamos na Prática 14. Com isso, podemos usar comandos mais simples para "ler" e "escrever" um *pixel* em uma posição (*i,j*) qualquer:

```
p = im[i][j]
...
im[i][j] = p
```

### Roteiro de Prática

Uma visão geral do programa (algoritmo de alto nível) segue abaixo:

1. Leitura do fator *f* entre 0 e 1
2. Leitura e exibição das duas imagens *im1* e *im2*
3. Sobreposição das imagens para dentro de *im3*
4. Exibição de *im3*
5. Cálculo e exibição do RMSE entre *im1* e *im3*
6. Cálculo e exibição do RMSE entre *im2* e *im3*

O algoritmo acima já está implementado dentro do esqueleto do programa. Para que ele funcione direito, você deverá preencher a implementação das seguintes funções:

**leiaFloat( msg, vmin, vmax )**

Essa função faz a leitura de um valor REAL entre **vmin** e **vmax** e retorna o valor lido. O parâmetro **msg** é usado para exibir a mensagem do input.

**sobrepoeImagens( im1, im2, im3, fator )**

Efetua a sobreposição das imagens **im1** e **im2**, considerando um **fator** entre 0 e 1, e coloca o resultado dentro da imagem **im3**.

Essa sobreposição será feita através de uma combinação linear entre cada *pixel* da imagem 1 e da imagem 2. Considere o *pixel* *p1* da imagem 1, e o *pixel* *p2* da imagem 2, ambos obtidos na mesma posição (*i,j*). Uma combinação linear entre *p1* e *p2* pode ser obtida como:

$$p3 = \text{round}( p1 * \text{fator} + p2 * (1 - \text{fator}) )$$

A função `round()` serve para arredondar o valor para o número inteiro mais próximo. O *pixel* p3 deve então ser colocado na mesma posição (*i,j*) da imagem **im3**. Ou seja:

```
im3[i][j] = round( im1[i][j] * fator + im2[i][j] * (1-fator) )
```

Para percorrer todas as posições (dos *pixels*) de uma imagem **img** qualquer, use o modelo já visto nos slides da Aula 8:

```
para i = 0 até img.altura-1:
    para j = 0 até img.largura-1:
        ...
```

A função deve considerar que a imagem **im3** já foi criada previamente, ou seja, ela só precisa ser preenchida como mostrado acima.

### RMSE( f, g )

Essa função deve calcular e retornar uma medida de erro chamada *Root Mean Square Error* (RMSE) entre duas imagens **f** e **g**. Esse cálculo é dado pela seguinte expressão:

$$RMSE = \sqrt{\frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (f[i][j] - g[i][j])^2}$$

onde *m* e *n* representam o número de linhas e colunas da imagem, respectivamente.

**Obs.:** a somatória dupla na equação acima apenas significa que, para cada valor de *i*, devemos percorrer todos os valores de *j*. Para traduzir isso em código Python, usamos um duplo **for**:

Notação matemática	Código em Python
$soma = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{ij}$	<pre>soma = 0 for i in range( 0, m ):     for j in range( 0, n ):         soma = soma + A[i][j]</pre>

Seguem abaixo três saídas produzidas pelo programa:

#### Exemplo 1:

```
Entre com o percentual para a imagem 1 (0..1): -1
Valor deve estar entre 0 e 1
Entre com o percentual para a imagem 1 (0..1): 1.1
Valor deve estar entre 0 e 1
Entre com o percentual para a imagem 1 (0..1): 0
Fazendo o "merge" das duas imagens...
RMSE entre as imagens 1 e 3: 112.9
RMSE entre as imagens 2 e 3: 0.0
```

#### Exemplo 2:

```
Entre com o percentual para a imagem 1 (0..1): 0.5
Fazendo o "merge" das duas imagens...
RMSE entre as imagens 1 e 3: 56.5
RMSE entre as imagens 2 e 3: 56.5
```

#### Exemplo 3:

```
Entre com o percentual para a imagem 1 (0..1): 0.9
Fazendo o "merge" das duas imagens...
RMSE entre as imagens 1 e 3: 11.3
RMSE entre as imagens 2 e 3: 101.7
```



Não esqueça de preencher o cabeçalho com seus dados e uma breve descrição do programa.

Após certificar-se que seu programa está correto, envie o arquivo do programa fonte (**p17.py**) através do sistema do LBI.