

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

## Introdução

Uma brincadeira comum no dia dos namorados é fazer a “brincadeira do espelho”, onde o casal se posiciona diante de um espelho e, através de um jogo de luzes, consegue obter uma imagem “fundida” entre os dois, como mostra o exemplo ao lado.

Nesta prática, você fará um programa para simular essa brincadeira. 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 de um valor  $a$  entre 1 e 100
2. Leitura de um valor  $d$  entre -100 e 100
3. Leitura e exibição das duas imagens  $im1$  e  $im2$
4. Merge das imagens para dentro de  $im3$
5. Exibição de  $im3$

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.

**f( x, a )**

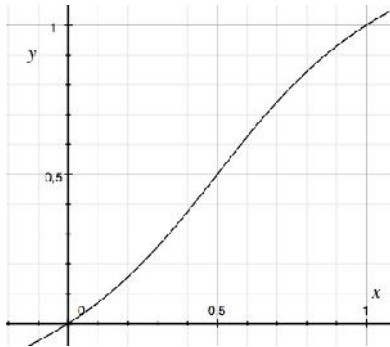
Essa função recebe como parâmetros os valores de  $x$  e de  $a$ , e retorna o valor da função:

$$y = 0,5 \left( 1 - \frac{\arctan(ax - a/2)}{\arctan(-a/2)} \right)$$

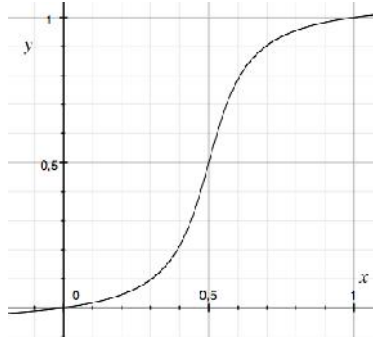
O arco tangente pode ser obtido com a função `math.atan()`.

A seguir temos o gráfico de  $y$  x  $x$  para essa função, para três valores da constante  $a$ .

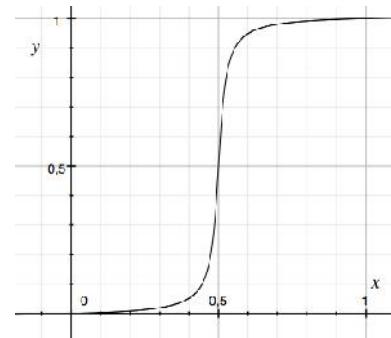
## Prática Substitutiva – INF100 – 2017/I – Valor: 4 pontos



(a)  $a = 2$



(b)  $a = 10$



(c)  $a = 50$

**mergeImagens( im1, im2, im3, a, d )**

Essa função deverá fazer a união (*merge*) entre as imagens **im1** e **im2**, usando os parâmetros **a** e **d**, e colocar o resultado dentro da imagem **im3**. A função deve considerar que a imagem **im3** já foi criada previamente, ou seja, ela só precisa ser preenchida.

Uma maneira bem simples de fazer isso seria copiar as duas metades para dentro da imagem **im3**. No entanto, isso produziria uma transição muito brusca entre as imagens. Para evitar isso, usaremos a função **f( x, a )** explicada acima.

Nos gráficos acima, usaremos o eixo  $x$  para representar a largura da imagem. O valor 0.5 representa a metade das duas imagens. O valor de  $y = f(x, a)$  será usado para multiplicar os valores dos *pixels* da imagem **im1**. O complemento desse valor ( $1 - y$ ) será usado para multiplicar os valores dos *pixels* da imagem **im2**.

Com isso, a junção propriamente dita das imagens será feita através de uma combinação linear entre cada *pixel* da imagem 1 e da imagem 2. Considere o *pixel*  $p_1$  da imagem 1, e o *pixel*  $p_2$  da imagem 2, ambos obtidos na mesma posição  $(i, j)$ . Uma combinação linear entre  $p_1$  e  $p_2$  pode ser obtido assim:

$$p_3 = \text{round}( p_1 * \text{fator} + p_2 * (1 - \text{fator}) )$$

onde **fator** = **f( x, a )**.

Desdobrando os pixels em suas representações matriciais, temos:

$$\text{im3}[i][j] = \text{round}( \text{im1}[i][j] * \text{fator} + \text{im2}[i][j] * (1 - \text{fator}) )$$

A função **round()** serve para arredondar o valor para o número inteiro mais próximo. O valor de **x** pode ser obtido como sendo:

$$x = (j + d) / \text{im1.largura}$$

onde **d** é o deslocamento horizontal com relação ao centro da imagem.

A imagem **im3** pode ser então obtida assim:

para  $i = 0$  até  $\text{im1.altura}-1$ :

para  $j = 0$  até  $\text{im1.largura}-1$ :

Calcule  $x$  conforme a fórmula acima

Calcule **fator** = **f( x, a )**

Calcule  $\text{im3}[i][j]$  conforme a combinação linear mostrada acima

Seguem abaixo os testes obrigatórios que você deverá fazer com o programa:

	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5
<b>a</b> =	5	50	100	100	100
<b>d</b> =	0	0	0	-100	100



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.