

**Obs. 1:** Os exercícios-programas devem ser resolvidos individualmente. Não serão aceitos EPs em grupo nem EPs iguais.

**Obs. 2:** Cada dia de atraso acarreta uma perda de 1 ponto no exercício.

**Obs. 3:** Procure enviar um único email para entregar o seu EP. Se você enviar dois ou mais emails, será considerado somente o email enviado por último.

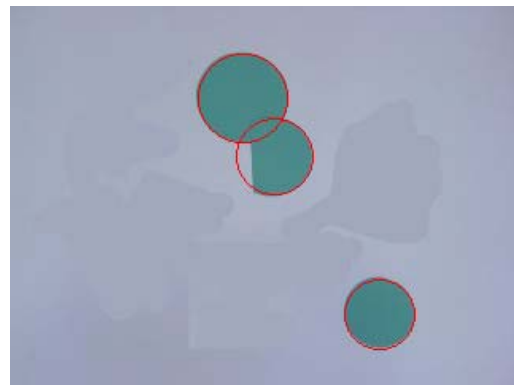
Faça um programa C/C++ chamado **ep3.cpp** que, dada uma imagem colorida **fig?.jpg** e dois números inteiros  $r_{min}$  e  $r_{max}$ , detecta todos os círculos (completos ou pedaços deles) com raios  $r$  tais que  $r_{min} \leq r \leq r_{max}$  (unidade em pixels). O comando-exemplo abaixo:

```
>EP3 fig1.jpg 20 30 fig1sai.txt fig1sai.tga [outros parâmetros]
```

deve ler **fig1.jpg**, detectar os círculos com raios entre 20 e 30 pixels, e gerar arquivos **fig1sai.txt** e **fig1sai.tga**. Você deve especificar o que são os “outros parâmetros”, de acordo com as necessidades do seu algoritmo.



**fig1.jpg**



**fig1sai.tga**

```
3
195 232 22
95 166 24
58 146 28
```

**fig1sai.txt**

A **fig1sai.txt** indica que há 3 círculos na imagem de entrada: ( $l=195$ ,  $c=232$ ,  $r=22$ ); ( $l=95$ ,  $c=166$ ,  $r=24$ ); e ( $l=58$ ,  $c=146$ ,  $r=28$ ). Esses círculos estão pintados de vermelho na imagem **fig1sai.tga**.

Obs. 1: Pode usar (ou não) a biblioteca-IMG.

Obs. 2: Teste o seu programa para as imagens **fig?.jpg**:

`http://www.lps.usp.br/~hae/psi5796-05/ep3/fig1.jpg`

`http://www.lps.usp.br/~hae/psi5796-05/ep3/fig2.jpg`

`http://www.lps.usp.br/~hae/psi5796-05/ep3/fig3.jpg`

`http://www.lps.usp.br/~hae/psi5796-05/ep3/fig4.jpg`

Obs. 3: O seguinte comando da biblioteca-IMG desenha o círculo vermelho ( $l=195$ ,  $c=232$ ,  $r=22$ ) na imagem colorida **a**:

`circulo(a,195,232,22,COR(255,0,0));`

Obs. 4: Entregue o arquivo-fonte **ep3.cpp** e um documento **coment.pdf** ou **coment.doc**. O documento **coment** deve conter obrigatoriamente as seguintes informações:

1. Os comandos e parâmetros utilizados para detectar os círculos de cada **fig?.jpg**. Estes comandos serão utilizados para verificar se o seu programa funciona corretamente. Exemplo:

`EP3 fig1.jpg 20 30 fig1sai.txt fig1sai.tga 0.6 0.3 100`

`EP3 fig2.jpg 15 35 fig2sai.txt fig2sai.tga 0.55 0.37 140`

`etc.`

2. Explicação do que significa cada um dos “outros parâmetros”.

3. Descrição do método que você implementou.

4. Nota: Não é necessário copiar o programa **ep3.cpp** para dentro do **coment**.

Obs. 5: Compacte os dois arquivos como **SeuNome\_Sobrenome.ZIP** e envie um email colocando como assunto “**PSI-5796 EP3**” para:

- **hae@lps.usp.br**

Obs. 6: O seu programa será corrigido compilado-o e testado-o em Windows, independentemente do sistema onde ele tiver sido originalmente escrito. Assim, se você fizer o seu programa em Linux (ou em qualquer outro sistema operacional diferente de Windows), convém testar se ele também funciona corretamente em Windows. Em quase todos os casos, um programa C++/IMG que funciona corretamente em Linux funcionará também em Windows (e vice-versa). Porém, há casos raros de programas que funcionam corretamente num sistema mas não no outro (como no caso de programas altamente recursivos).