

- Trabalho Prático -

Grupo: 3 pessoas

Entrega: 05/12/2022

Valor: 10 pontos

Linguagem de programação: C



PUC Minas

Neste trabalho prático você irá demonstrar suas habilidades em análise e programação. Como tema, será abordado a manipulação de matrizes utilizando imagens. Em referido trabalho, dentre outras, você irá expor suas habilidades de trabalhar em equipe, e aplicar conhecimentos sobre manipulação de variáveis, estruturas de dados, funções, passagens de parâmetros, arquivos e modularização.

De maneira resumida, seu algoritmo deverá ler um arquivo de imagem do tipo "Portable Gray Map" (PGMA). PGMA é um formato de dados que contém imagens em tons de cinza. Como exemplo:



A definição do arquivo para a imagem acima é a seguinte:

```
P2
# feep.ascii.pgm
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

- A extensão de arquivo típica é ".pgm", mas uma extensão de ".pnm" também é usada ocasionalmente;
- Um "identificador mágico" para identificar o tipo de arquivo. O identificador mágico de um arquivo ASCII PGM são os dois caracteres "P2";
- Caracteres de um "#" até o próximo final de linha são ignorados (comentários);
- A largura, um espaço em branco e a altura da imagem;
- O valor máximo de cinza (tonalidade);
- "**n**" linhas, onde cada linha contém **m** tonalidades, cada tonalidade representa 1 a cor de um pixel.

Algumas versões do *windows* não oferecem suporte nativo para a visualização deste tipo de imagem, assim, você pode instalar, dentre outras opções:

- IrfanView – visualizador leve, pode ser baixado em <https://www.solvusoft.com/en/file-extensions/software/open-source/irfanview/>
- GIMP – editor de imagens mais pesado, pode ser baixado em <https://www.solvusoft.com/en/file-extensions/software/the-gimp-development-team/gimp/>

Seu programa deverá ler o arquivo, e salvar a imagem em uma matriz alocada dinamicamente. Lembre-se de liberar a memória usada ao final do programa. Seu programa deverá gerar diversas alterações na imagem, e para cada alteração, um arquivo com a imagem alterada deverá ser gerado. Ao gerar o arquivo com a imagem alterada, tenha cuidado para não alterar o formato da imagem, como mostrado na figura acima.

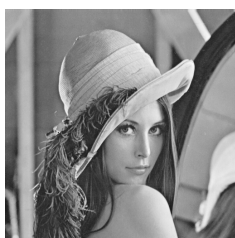
As ações que devem ser implementadas são:

1 - Leitura da imagem e alocação em uma matriz. Seu programa deve funcionar para diferentes imagens.

Exemplo de imagem de entrada: “lena.pgm”



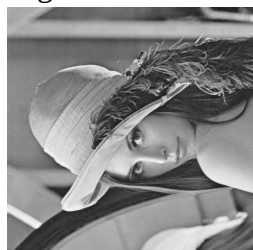
1 – Geração de uma copia idêntica a imagem original.



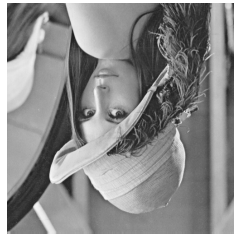
2 – Saída rotacionada em 90 graus de modo normal e espelhado



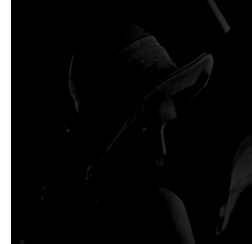
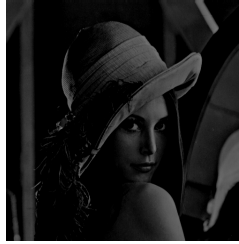
3 – Saída rotacionada em 270 graus de modo normal e espelhado



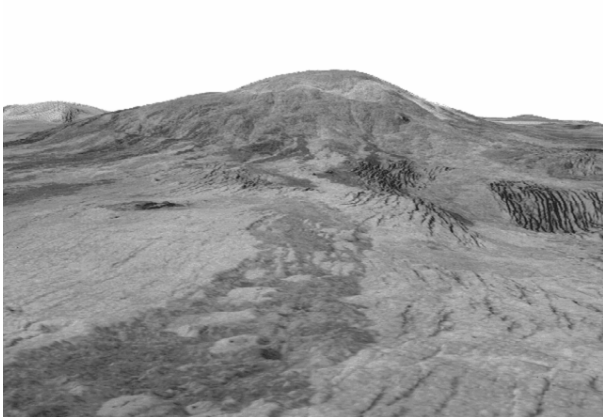
4 – Saída rotacionada em 180 graus de modo normal e espelhado



5 – Saída escurecida em tonalidades ex: escurecida em 25%, 50% e 75%.



Outras imagens de entrada serão passadas em um arquivo pelo *Canvas*. Seu algoritmo deve funcionar para todas.



Este trabalho deve adotar boas praticas de programação, como código modularizado, comentado, limpo e bem organizado. O programa devera ser feito em uma máquina local em um ambiente como o Code::Blocks. Outros serão aceitos, desde que o aluno saiba como usar.

Um esqueleto da implementação, com algumas funções, pode ser acessado em <https://onlinegdb.com/21dn-inR2>

Avaliação:

O código deve ser entregue em um arquivo somente, no formato “main.c”. Será avaliada a organização, clareza e comentários em todo o código. Cada função, bem como cada parâmetro deverá ser documentado através de comentários.

Será feita uma entrevista com os alunos do grupo, sobre o código e seu funcionamento. As notas serão individuais, se algum componente não participar, ou demonstrar na entrevista que não conhece o funcionamento do programa, este poderá perder pontos. O aluno que não participar da entrevista receberá nota 0;

Etapas do trabalho:

- Implementação;
- Explicação da implementação;

Implementação

Os algoritmos deverá ser implementados na linguagem C, em um computador físico (**não** usar o GBD online), de forma modularizada com **comentários**, utilizado funções, passagem de parâmetros e **alocação dinâmica de memória**. Códigos que não funcionam, não ou pouco comentados, e práticas indevidas de programação, como nomear errado as variáveis e funções, acarretarão em desconto de nota.

No caso de dúvidas, pergunte o professor :)

Um abraço, e adiantem-se, quanto mais cedo começarem, mais cedo vão terminar!