INSTITUTO FEDERAL Brasília	**Instituto Federal de Brasília** **Campus Taguatinga** **Superior em Computação** **Especificação do Trabalho 2** **Biblioteca Simplificada de Imagens PPM** **Programação de Computadores 1** **Prof. João Victor de A. Oliveira**

Data de Entrega: 01/08/2025

Pode ser feito em dupla!

Formato de entrega: Pasta zipada, nomeada como T1_nome1_nome2.zip, contendo todos os códigos fonte necessários para executar o programa e um arquivo **README.txt** contendo os nomes completos dos integrantes, as instruções de compilação.

Obs.: Deve-se usar C padrão para a criação desta biblioteca (são permitidas o uso de funções do padrão POSIX, desde que sejam descritas no arquivo README.txt); o programa será testado em ambiente linux.

Criação de uma Biblioteca de Imagens no Formato PPM

O formato de arquivo ppm (*Portable Pixel Map)* é utilizado para armazenar imagens. Sua especificação é bastante simples, porém ineficiente e altamente redundante, contendo diversas informações que um ser humano não notaria em uma imagem. Por sua simplicidade, este formato pode ser usado para escrever e analisar programas de manipulação de imagem de forma simples e prática.

Neste trabalho iremos desenvolver uma biblioteca de imagens no formato PPM que possuirá as seguintes funcionalidades:

- Redimensionamento de imagem;
- Rotação de imagem;
- Conversão para Escala de cinza;
- Inversão de cores (efeito negativo);

Nas próximas seções desta especificação será apresentado o formato de arquivo PPM e dado detalhes das funcionalidades esperadas na biblioteca e considerações finais.

1. Especificação do formato de arquivo .ppm

Cada imagem em formato .ppm consiste das seguintes informações (em ordem):

1. Um *magic number,* usado para identificar um arquivo de extensão ppm, consistindo de dois caracteres: “**P6”;**
2. Um caractere *whitespace* (espaço em branco, tabulações, quebra de linhas \r, \n, …);
3. Uma largura (*width)*, formatada usando uma representação decimal com caracteres ASCII;
4. Um caractere whitespace;
5. Uma altura (*height),* formatada usando uma representação decimal com caracteres ASCII;
6. um caractere *whitespace*;
7. Um valor máximo (**maxval**) de uma cor em ASCII decimal. Este valor deve ser menor que 65536 e maior que zero;
8. Um caractere *whitespace* (normalmente uma quebra de linha);
9. Uma matriz, onde teremos **n** linhas e **m** colunas, onde **n** é a altura e **m** a largura da imagem. Essa matriz tem em cada elemento um pixel que pode ser definido com três valores (RGB):
    a. Vermelho: 1 byte, se **maxval** < 256, ou dois bytes, caso contrário;
    b. Verde: 1 byte, se **maxval** < 256, ou dois bytes, caso contrário;
    c. Azul: 1 byte, se **maxval** < 256, ou dois bytes, caso contrário;

Para mais informações sobre o padrão, acesse sua especificação em: http://netpbm.sourceforge.net/doc/ppm.html

Obs.: para este trabalho assumimos que todas as imagens possuem maxval igual a 255.

2. Funcionalidades esperadas na biblioteca de imagem

A biblioteca deve ser criada em um módulo chamado **imagens_ppm** e deve possuir um arquivo de cabeçalho (.h) e um arquivo de implementação (.c). Cada funcionalidade deve possuir uma função própria, com seu protótipo especificado no arquivo cabeçalho.

De forma a testar a biblioteca, deve ser criado um programa principal **(main.c)** que irá fornecer a possibilidade do usuário testar a biblioteca de imagens. Este programa principal deve receber via linha de comando a imagem a ser testada, a transformação a ser feita nessa imagem e o nome de um arquivo de saída.

Ex.: Ao executar o comando**:**

./*main* sonic.ppm rotacao 90 sonic_output.ppm

A imagem de entrada sonic.ppm será rotacionada em 90 graus e salva no arquivo sonic_output.ppm.

A seguir são descritas o que deve ser feito em cada uma das funcionalidades esperadas nesta biblioteca.

2.1 Redimensionamento de imagem

Uma imagem pode ser ampliada ou reduzida de acordo com uma quantidade de vezes pré-determinada. Para realizar um redimensionamento o usuário digitará o seguinte comando:

./main entrada.ppm [amp|red] X saida.ppm

onde entrada.ppm e saida.ppm são os arquivos de imagem de entrada e saída, respectivamente, já **amp** indica que a imagem deve ser ampliada em X vezes e **red** de que a imagem deve ser reduzida em X vezes. Por fim, **X** é um número inteiro que varia de 1 a 4 (indicando quantas vezes mais a imagem será ampliada ou reduzida).

Ex.: Seja a imagem entrada.ppm com dimensões 100x100:

./main entrada.ppm amp 3 saida.ppm

saida.ppm será gerada com dimensões 300x300.

2.2 Rotação de imagem

Uma imagem poderá ser rotacionada em ângulos retos (90º, 180º e 270º). Para realizar a rotação o usuário deverá digitar o seguinte comando:

./main entrada.ppm rot X saida.ppm

onde **rot** indica a funcionalidade de rotação e X indicará qual o ângulo de rotação.

2.3 Conversão para Escala de cinza

Deverá ser possível realizar uma conversão de uma imagem colorida para a escala de cinza. Para fazer isso, deve-se aproximar cada pixel da imagem à cor em tom de cinza mais próxima, de acordo com o seu código RGB. Neste trabalho usaremos uma escala de 10 tons de cinza, conforme a tabela abaixo:

Nome da Cor	Código RGB
Black	(0,0,0)
grey11	(28,28,28)
grey21	(54,54,54)
grey31	(79,79,79)

DimGray	(105,105,105)
Gray	(128,128,128)
DarkGray	(169,169,169)
Silver	(192,192,192)
LightGrey	(211,211,211)
Gainsboro	(220,220,220)

fonte: https://celke.com.br/artigo/tabela-de-cores-html-nome-hexadecimal-rgb

Para converter a imagem em escala de cinza o usuário deverá digitar o seguinte comando:

./main entrada.ppm cinza saida.ppm

Onde **cinza** indica a funcionalidade de conversão para a escala de cinza.

2.4 Inversão de cores (efeito negativo)

Outra funcionalidade esperada é a inversão de cores, ou efeito negativo, onde cada pixel terá sua cor invertida. Note que no formato RGB, cada pixel pode ser representado por um número que varia de 0 (preto) a 255 (branco). Logo, para inverter as cores, basta pegar o maior valor, ou seja 255, e decrementar pelo valor de cada cor básica (R, G e B).

Ex.: seja a cor RGB = (50,100,200), ou seja R=50, G=100 e B=200. 

A cor invertida será:

R = 255 - 50 = 205;
G = 255 - 100 = 155;
B = 255 - 200 = 55;

RGB invertido = (205,155,55). 

Para inverter as cores, o usuário deverá digitar o seguinte comando:

./main entrada.ppm inverte saída.ppm

Considerações Finais

Para converter uma imagem de outra extensão para o formato .ppm, pode-se utilizar o comando (linux):

convert imagem.jpg imagem.ppm

Como qualquer projeto de software, qualquer dúvida na especificação deve ser esclarecida com a parte interessada (no caso o professor 🙂)