Universidade de São Paulo – USP
Instituto de Ciências Matemáticas e de Computação – ICMC
Departamento de Ciências de Computação – SCC

SCE-503 - Algoritmos e Estruturas de Dados II

Responsável: Prof. Gustavo Batista
gbatista@icmc.usp.br

Projeto - Arquivos – ASCIIArt

Data de Entrega – 30/06

1. Objetivo

Escrever um programa que recebe uma imagem BMP como entrada e gera uma imagem semelhante em caracteres ASCII.

2. Especificações do formato de arquivo binário BMP

Um arquivo BMP é composto de um cabeçalho (*header*), uma tabela de cores (que guarda os valores RGB de cores que serão referenciadas futuramente apenas pelos seus índices na tabela de cores) e o finalmente as informações dos *pixels* da imagem, começando do canto inferior esquerdo até o canto superior direito.

2.1 BMP file *header*

Neste projeto, por simplicidade, considere que a imagem não é indexada, ou seja, não possui tabela de cores e assim a cada pixel é associado diretamente um valor RGB (ou RGBA, para imagens com transparência). As linhas destacadas em negrito devem ser tratadas pelo programa. As outras podem ser lidas e descartadas.

O formato do *header* está especificado abaixo:

Offset	Tamanho	Propósito
0	**2**	**utilizado para identificar o arquivo BMP: 0x42 0x4D (código ASCII para as letras B e M)**
2	4	o tamanho do arquivo BMP em bytes
6	4	reservado; valor atual depende da aplicação que gerou a imagem
10	4	o offset, i. e, o endereço inicial a partir do qual as informações dos *pixels* podem ser encontradas
14	4	o tamanho deste cabeçalho (40 bytes)
18	**4**	**a largura da imagem em *pixels* (*signed integer*)**
22	**4**	**a altura da imagem em pixels (*signed integer*)**
26	2	o número de camadas de cores utilizado. Deve ser fixado em 1

28	**2**	**o número de bits por pixel. Valores típicos são: 24 e 32**
30	4	o método de compressão utilizado
34	4	o tamanho da imagem (tamanho das informações sobre os *pixels*) em bytes
38	4	a resolução horizontal da imagem (*pixels* por metro, *signed integer*)
42	4	a resolução vertical da imagem (*pixels* por metro, *signed integer*)
46	4	o número de cores na palheta de cores, ou 0 para utilizar o valor *default*: 2*n*
50	4	o número de cores importantes utilizadas, ou 0 quando toda cor é importante. Campo geralmente ignorado

2.2 Especificação do trabalho

A idéia geral do trabalho é transformar uma imagem em formato BMP fornecida para o programa em um conjunto de caracteres ascii no qual um grupo de *pixels* de tamanho *M*x*N*, (*M* e *N* devem ser constantes do programa) são convertidos em 1 caractere. Para isso, se associa cores mais escuras à caracteres mais "cheios" e cores mais claras à caracteres mais "vazios" (ver exemplo no final do enunciado deste trabalho). Uma sugestão inicial de caracteres, em ordem de cheio para vazio seria:

```
{'#','$','O','=','+','|','-','^','.',' '}
```

Após a leitura do *header*, faça um loop duplo para a altura e largura da imagem, leia as informações dos *pixels* (verifique a quantidade de bits por pixel no *header*) e coloque estas informações em uma matriz. Caso a quantidade de *bits* por pixel seja 32, deve-se descartar os últimos 8 *bits* (*bits* de transparência). Pode-se guardar apenas uma média dos valores RGB na matriz.

Dica: para a leitura das informações dos *pixels*, utilize um vetor de caracteres e faça um *type cast* para *short* em um elemento do vetor para obter um valor entre 0 e 255. Como exemplo, o trecho de código abaixo lê as informações de um pixel e coloca na variável inteira *color* a média dos valores RGB do pixel.

```
unsigned char cbuf[4];
fread(cbuf,1,bitsperpixel/8,infile);
int color = ((short)cbuf[2]+(short)cbuf[1]+(short)cbuf[0])/3;
```

Percorra blocos de tamanho *M*x*N* desta matriz e decida qual caractere deve representar aquele bloco (associe um caractere mais vazio para blocos claros e um caractere mais cheio para blocos escuros). A heurística mais simples para decidir se um bloco é claro ou escuro é fazer uma média dos valores dos *pixels* do bloco.

Imprima os caracteres na saída padrão, do canto superior esquerdo para o canto inferior direito (observe que, no formato BMP, esta ordem não é a mesma: as informações dos *pixels* são dadas do canto inferior esquerdo ao canto superior direito).

2.3 Testes

Teste seu programa com a imagem socrates.bmp para M = 1 e N =2.

Mude o vetor de *shades* a gosto e teste novamente seu programa.

Teste para outras imagens BMP. Caso o programa não funcione, verifique o porquê (se há algum valor importante do header que deixamos de ler, ou se a imagem utiliza compressão, palheta de cores, etc.).

2.4 Entregáveis

Os entregáveis para este projeto são o código fonte e um relatório curto de até 3 páginas descrevendo a solução, dificuldades e melhoramentos realizados. O relatório deve ter os nomes de todos os membros do grupo, sendo composto, no máximo, por três integrantes.

2.5 Exemplo: socrates.bmp