

Alan Ludke - 17205070
Felipe Santos - 17200441
Franck Kumako - 17150563

Relatório

Responda as questões e entregue o código modificado:

Questão 1. Abra o arquivo lena.bmp no editor hexadecimal em <https://hexed.it/> e indique no relatório:

Qual é o valor do campo offset?

R: Segundo o item 2 (Formato BMP), o offset está presente nos bytes 10-13. Ela indica a posição onde termina o cabeçalho e começa a parte de dados da imagem.

lena.bmp x																
00000000	42	4D	36	00	0C	00	00	00	00	00	36	00	00	00	28	00
00000010	00	00	00	02	00	00	00	02	00	00	01	00	18	00	00	00
00000020	00	00	00	00	00	00	0C	00	00	00	0C	00	00	00	0C	00
00000030	00	00	00	00	00	00	39	16	52	39	16	52	3E	20	60	3E

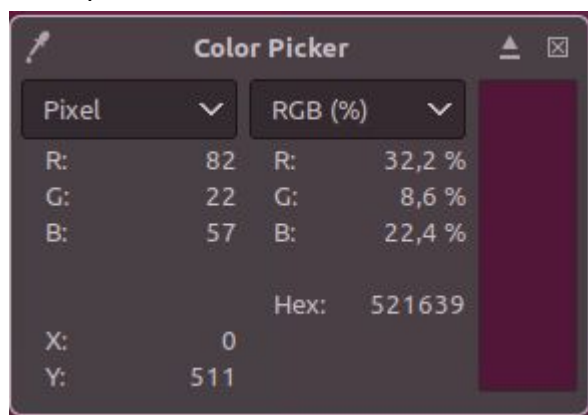
36 00 00 00

Qual é o tamanho de bytes do cabeçalho deste arquivo?

R: De acordo com a tabela 1 o cabeçalho de um arquivo bmp tem 54 bytes

Quais são os valores dos componentes de cor (RGB) do primeiro pixel armazenado no arquivo (primeiro pixel da última linha da imagem quando ela for visualizada)?

R: De acordo com o Gimp, os valores de RGB do primeiro pixel da última linha são 82, 22 e 57 respectivamente.



e pelo HexEd podemos confirmar isso, vemos que o primeiro pixel da imagem (no offset 54) tem os valores 39 16 52 (hex), e pelo roteiro vemos que o BPM organiza os pixels RGB definindo os valores de B, G e depois R, ou seja

Blue -> 39 (hex) = 57 (10)

Green -> 16 (hex) = 22 (10)

Red -> 52 (hex) = 82 (10)

42	4D	36	00	0C	00	00	00	00	00	36	00	00	00	28	00
00	00	00	02	00	00	00	02	00	00	01	00	18	00	00	00
00	00	00	00	00	00	00	0C	00	00	00	0C	00	00	00	00
00	00	00	00	00	00	39	16	52	39	16	52	3E	20	60	3E
1C	5D	41	1E	61	39	15	5E	3E	1A	5C	3C	1C	5D	44	24

Questão 2. Corrigir o erro no código para que a imagem decodificada seja igual a imagem original. O código corrigido deve ser entregue. Dica: a implementação de tal conversão está no arquivo Bitmap.java, método cuif1toRaster (linha 219). Também pode analisar o conteúdo do arquivo original BMP e aquele decodificado, por exemplo usando <https://hexed.it/>

R: Como a conversão utiliza BGR ao invés de RGB foi-se necessário modificar a linha 255, 256 e 257. Por isso, B passa a ser no index 0 e R no índice 2.

```

251 // Escrita no array de bytes (bitmapfile) que representa em memória o arquivo Bitmap
252 int k=offset to start image;
253 for (int i=0; i<height; i++){
254     for (int j=0; j<width; j++){
255         bitmapfile[k++] = (byte)raster[i][j][2]; // escreve componente R
256         bitmapfile[k++] = (byte)raster[i][j][1]; // escreve componente G
257         bitmapfile[k++] = (byte)raster[i][j][0]; // escreve componente B
258     }
259 }
260

```

Questão 3. Qual é o tamanho do cabeçalho do arquivo lena.cuif?

R: 12 bytes de cabeçalho

Questão 4. Há perdas nos dados da imagem na conversão bmp → cuif (CUI.1) → bmp? Expliquem.

R: Não há perda nos dados da imagem na conversão pois as principais diferenças entre BMP e CUI.1 é a ordem dos canais RGB e a maneira de codificação. No BMP a codificação é em BGR enquanto em CUI.1 a codificação é em RGB. No BMP a codificação é feita pixel-a-pixel enquanto no CUI.1 a codificação é feita canal-a-canal. Além disso o tamanho dos arquivos são iguais. Ou seja, não há alteração na quantidade de dados.

Questão 5. Compacte separadamente a imagem lena.bmp e lena.cuif usando zip ou rar. Qual é o arquivo que compactou mais? Em seguida, indique a vantagem de organizar os pixels nesta sequência (primeiro os valores de R, depois de G e finalmente de B) para a compressão baseada em entropia, por exemplo, DPCM? Dica: lembre primeiro o princípio do DPCM e compare a parte de dados de imagem do arquivo lena.bmp e lena.cuif no editor hexadecimal.

R:

lena.bmp e lenadecodificada.bmp tem 786.5kB

lena.bmp.zip agora tem 715.6kB

lenadecodificada.bmp.zip agora tem 507.4kB

lenadecodificada compactou mais!

Se os pixels tiverem valores muito semelhantes, é possível usar um número menor de bits para armazenar o erro do que aquele usado para codificar o valor absoluto. Ao organizar da

seguinte maneira: primeiro todos os valores de R, depois os de G e depois os de B, podemos considerar que haverá valores mais próximos, devido à semelhança entre os valores das cores. Isso não seria tão viável se fosse codificado em partes RGB, pois poderia acarretar em um erro acumulado maior.

