UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CIÊNCIAS DA COMPUTAÇÃO

Gabriel Medeiros Lopes Carneiro
Lorenzo Lima Franco Maturano

Prática III

Florianópolis, SC
2021

1. Abra o arquivo lena.bmp no editor hexadecimal em https://hexed.it/ e, analisando o formato do cabeçalho BMP apresentado sa Seção 2, indique no relatório: qual é o valor dos campos *offset* e *tamanho do arquivo*? Qual é o tamanho do cabeçalho deste arquivo? Quais são os valores dos componentes de cor (RGB) do segundo pixel armazenado no arquivo (na visualização da imagem, seria segundo píxel da última linha da imagem)?

   - Offset: 0x36000000.
   - Tamanho do arquivo: 786486 bytes, de acordo com o hexed.
   - Tamanho do cabeçalho: 54 bytes.
   - Valores RGB:

2. Corrigir é o erro no código para que a imagem decodificada seja igual a imagem original. O código corrigido deve ser entregue. Dica: o erro está na operação de converter a parte de dados (raster) da imagem CUIF.1 para a parte de dados (raster) do arquivo BMP que está no arquivoBitmap.java, método cuif1toRaster(linha 221).

   Feito.

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

   12 bytes, se não considerarmos os identificadores dos alunos como parte do cabeçalho. 20 bytes, caso seja considerado.

4. Há perdas nos dados da imagem na conversãobmp → cuif(CUI.1) → bmp? Justifique sua resposta.

   Não. O que foi feito é representar a mesma informação em outro formato (BGR ou RGB), sem compressão. Os arquivos possuem até o mesmo tamanho.

5. Qual é a vantagem de organizar os pixels nesta sequência definida pelo CUIF.1 (primeiro os valores de R, depois de G e finalmente de B) para a compressão baseada em RLE ou DPCM? Dica: relembre os princípios da compressão RLE e DPCM e compare a parte de dados de imagem do arquivo lena.bmp e lena.cuif no editor hexadecimal.

   Como no CUIF cada canal RGB completos em sequência, ao contrário do BMP onde cada pixel possui um canal BGR, há mais chance de se ter valores iguais ou semelhantes próximos. Geralmente pixels vizinhos tem alta taxa de semelhança. Isso é excelente para compressão baseada nesses métodos.