

Construa os programas indicados usando a linguagem C. Entregue o código desenvolvido, devidamente indentado e comentado, e um relatório com a descrição das soluções. Inclua na entrega o *makefile* que permita gerar os ficheiros executáveis a partir do código fonte. O relatório deverá ser um guia para a compreensão do código desenvolvido e não uma mera tradução deste para língua natural. Contacte o docente se tiver dúvidas. Encoraja-se a discussão de problemas e soluções com colegas de outros grupos, mas recorda-se que a partilha directa de soluções leva, no mínimo, à anulação das entregas de todos os envolvidos.

Os formatos de imagem PBM, PGM e PPM [<http://ozviz.wasp.uwa.edu.au/~pbourke/dataformats/ppm/>] permitem o armazenamento de imagens a preto e branco, escala de cinzentos e a cores, respectivamente. Em anexo é fornecido um [conjunto de imagens](#) para usar nos exercícios propostos. A este conjunto pode juntar outras imagens nestes formatos.

1. Escreva um programa (*imgInfo*) que apresente no ecrã a dimensão (*largura* x *altura*), o tipo de representação (*P1*, *P2*, ..., *P6*) e o valor máximo admissível para cada campo de informação sobre os *pixels* da imagem armazenada num ficheiro PBM, PGM ou PPM. Indique também qual o ponto no ficheiro em que começa a informação sobre os *pixels* da imagem. Apresente no relatório o resultado deste programa em todas as imagens do conjunto.
2. A biblioteca *zlib* [<http://www.zlib.net/>] fornece funções que realizam a compactação (*deflate*) e descompactação (*inflate*) de dados baseados numa variante do algoritmo LZ77. Utilizando a informação disponível no documento *online* “*zlib Usage Example*”, gere o compactador/descompactador (*zpipe*) e compacte as imagens PPM do conjunto, apresentando no relatório as dimensões (em *bytes*) dos ficheiros originais e as do resultado da compactação.

O formato de imagem PNG [<http://www.libpng.org/pub/png/>] utiliza uma técnica de compressão, sem perdas (compactação), composta por dois passos: primeiro aplica um filtro de transformação sobre os *bytes* que descrevem os *pixels* da imagem original; o resultado da transformação é, em seguida, passado à *zlib* para compactação. Na especificação PNG são indicados 5 filtros diferentes (*None*, *Sub*, *Up*, *Average* e *Paeth*). No filtro *Sub*, por exemplo, cada *byte* de um *pixel* é transformado na diferença para o *byte* correspondente do *pixel* à sua esquerda.

3. Inspirando-se na técnica de compactação do formato PNG, construa um compactador (*ppmZip*) de imagens PPM P6 apenas com campos até 8 bits. Os ficheiros gerados devem ter extensão PPZ. Compacte as imagens do conjunto, apresentando no relatório as dimensões (em *bytes*) dos ficheiros originais e as do resultado da compactação. Apresente os resultados em conjunto com os obtidos no exercício 2.
4. Construa um descompactador (*ppmUnzip*) para reconstituir as imagens PPM P6 originais a partir dos ficheiros produzidos no exercício anterior. Reconstitua as imagens do conjunto a partir dos ficheiros PPZ gerados pelo compactador. Verifique que os ficheiros reconstituídos e os originais são iguais.

Valorizam-se os trabalhos organizados em módulos que partilhem código entre os vários exercícios e que obtenham melhores taxas de compactação.

Data limite de entrega: 2 de Dezembro de 2009

*Bom trabalho!*