

**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**  
**LICENCIATURA EM ENGENHARIA INFORMÁTICA E DE COMPUTADORES**  
**SISTEMAS MULTIMÉDIA**

**Primeiro Trabalho Prático - semestre de verão 2015/2016 (6 de abril de 2016)**

**Data Limite de Entrega (Código e Relatório): 9 de maio de 2016**

- 
1. Exercício 4 do guia de laboratório da primeira aula prática. Apresente todo o código desenvolvido, os resultados experimentais obtidos e os respetivos comentários.
  2. Pretende-se avaliar o desempenho de diferentes pares codificador/descodificador, sobre ficheiros normalizados de teste.
    - a) Escreva a função `coder_decoder_evaluator.m`, a qual complementa a função `coder_evaluator.m`, com: a respetiva descodificação, verificação de igualdade entre ficheiro original e descodificado, medição do tempo de codificação e de descodificação. Os codificadores a considerar são Huffman semi-adaptativo, Huffman adaptativo e aritmético semi-adaptativo (disponíveis no ficheiro `testCodersSM.zip`).
    - b) Escolha 12 ficheiros de teste (com 4 ficheiros de cada conjunto Calgary Corpus, Canterbury Corpus e Silesia Corpus). Indique os ficheiros escolhidos. Aplique a função `coder_decoder_evaluator.m` sobre os 12 ficheiros. Compare e comente os resultados obtidos, para os três pares codificador/descodificador.
    - c) Sobre os 12 ficheiros escolhidos na alínea anterior: aplique um codificador à sua escolha (e.g. 7-Zip, WinZip ou WinRar) ou o armazenamento através de `zipstream` (em Java ou C#). Indique o codificador/técnica escolhidos e apresente a taxa de compressão obtida para cada ficheiro. Compare as taxas de compressão com as obtidas na alínea b). Comente os resultados.
  3. Implemente uma fonte de símbolos com alfabeto e *função massa de probabilidade* (FMP) genéricos. As sequências produzidas pela fonte, com dimensão  $L$ , deverão ser escritas num ficheiro de texto (uma sequência por linha). Descreva a metodologia de geração de símbolos pela fonte. Apresente resultados de geração de sequências nas seguintes condições:
    - a) Sequências de algarismos decimais, com  $L=8$ , em que para o primeiro dígito  $d \in \{1, \dots, 9\}$ , se observa a Lei de Benford dada por  $p(d) = \log_{10} \left(1 + \frac{1}{d}\right)$ . Para os restantes dígitos, assume-se distribuição uniforme. Por exemplo, a sequência 10548723 poderá ser gerada, nesta situação.
    - b) Sequências alfanuméricas (algarismos decimais e letras maiúsculas), com  $L = 24$ , semelhantes a chaves de ativação e registo de software (por exemplo: RTY9 GHUI 1JER 82TY SGJP IUDS).
  4. Para teste de um sistema de informação associado a um jogo de sorte, pretende-se popular duas tabelas com dados, gerados aleatoriamente, sobre indivíduos (apostadores) e as respetivas apostas. Por cada indivíduo, teremos a seguinte informação:

Número de Cidadão | Nome(s) Próprio(s) e Apelido(s) | Concelho de Residência | Profissão

A tabela com as apostas terá registos na forma

Número de Cidadão | Aposta | Data

O número de cidadão (8 algarismos decimais) é único e podem existir nomes iguais. Recorra aos ficheiros `Nomes.txt`, `Apelidos.txt`, `Concelhos.txt` e `Profissões.txt` para gerar os diferentes campos do registo. Para a geração do nome completo, considere: um nome próprio e um apelido, dois nomes próprios e um apelido, dois nomes próprios e dois apelidos. A aposta é formada por 5 números inteiros (diferentes) no intervalo  $\{1, \dots, 50\}$  seguidos de outros 2 números inteiros (diferentes) no intervalo  $\{1, \dots, 11\}$ . A data tem a forma `dd-mm-aaaa`.

Estabeleça uma metodologia de preenchimento automático das duas tabelas. Descreva a metodologia, implemente-a e apresente resultados experimentais na forma de tabelas preenchidas, em ficheiros de texto, com mais de 1000 registos. Deverão existir indivíduos com 1, 2, 3 e 4 apostas; deverão também existir indivíduos que não realizam qualquer aposta.
  5. Escreva a aplicação `LZ78Tokenizer`, a qual realiza a decomposição LZ78 de um ficheiro de entrada. O dicionário inicial tem dimensão de 512 entradas, ficando a respetiva política de gestão à sua escolha. Os *tokens* LZ78 são escritos num ficheiro de texto de saída (um *token* por linha). No final do ficheiro de saída, deverá constar o conteúdo do dicionário final e o valor da taxa de compressão obtida pelo algoritmo LZ78. Apresente resultados experimentais obtidos com diferentes ficheiros e os respetivos comentários.