

Aula prática 1

Estes exercícios têm como primeiro objetivo aplicar conhecimentos de representação de dados e de ficheiros apresentados ainda em Programação 1 e na primeira aula teórica de Programação 2.

1 – Pretende-se obter estatísticas de ocorrências de letras em textos na língua portuguesa.

1.1 - Desenvolva um programa que leia um ficheiro de texto e permita obter médias de ocorrências de todas as letras do alfabeto. Tenha em conta que todas as letras devem ser convertidas para o mesmo caso (maiúscula ou minúscula), todas as vogais acentuadas devem ser convertidas para não acentuadas e todas as cedilhas devem ser retiradas dos “cês” com cedilha.

Utilize o ficheiro de texto *lusiadas.txt* para testar o programa que desenvolveu. O ficheiro está codificado no formato ISO/IEC 8859-1 (Latin-1) e deverá por isso usar a seguinte tabela de conversão:

letra	valor decimal	valor hex	letra	valor decimal	valor hex	letra	valor decimal	valor hex
À	192	C0	Ó	211	D3	É	201	C9
Á	193	C1	Ô	212	D4	Ê	202	CA
Â	194	C2	Õ	213	D5	é	233	E9
Ã	195	C3	ó	243	F3	ê	234	EA
à	224	E0	ô	244	F4	Ú	218	DA
á	225	E1	õ	245	F5	ú	250	FA
â	226	E2	í	205	CD	Ç	199	C7
ã	227	E3	í	237	ED	ç	231	E7

Exemplo usando o ficheiro *lusiadas.txt*

```
Numero de letras no ficheiro 246997
Letra a = 13.89%
Letra b = 0.97%
Letra c = 3.21%
...
Letra x = 0.15%
Letra y = 0.00%
Letra z = 0.37%
```

Sugestão: utilize a função `getc()`, guardando o resultado numa variável do tipo `int`. Depois compare com os valores decimais ou hexadecimais indicados na tabela.

1.2 – Considere que o mesmo documento foi guardado em formato UTF-8 no ficheiro com o nome *lusiadas-utf8.txt*. Compare o tamanho dos dois ficheiros e justifique a diferença.

1.3 – Desenvolva um novo programa que conte quantas vezes aparecem as seguintes letras do ficheiro em formato UTF-8:

letra	valor hex (2 bytes)	letra	valor hex (2 bytes)	letra	valor hex (2 bytes)
Ã	C3 83	Õ	C3 95	Ç	C3 87
ã	C3 A3	õ	C3 B5	ç	C3 A7

Exemplo usando o ficheiro *lusiadas-utf8.txt*

```
Letra A com til = 1620
Letra O com til = 115
Letra C com cedilha = 863
```

2 – Os ficheiros áudio MP3 incorporam, para além da informação áudio, informação associada normalmente designada de *metadata*; esta informação pode incluir, entre outros, dados sobre o nome do artista, da faixa e do álbum, comentários, etc. Os dois formatos de *metadata* em MP3 são o ID3v1 e ID3v2.

Escreva um programa que leia a *metadata* de ficheiros MP3 no formato ID3v1. Quando essa informação está presente, os campos de informação encontram-se no fim do ficheiro. São utilizados no total 128 bytes, sendo inicializados por 3 bytes contendo o valor “TAG”. A tabela em baixo resume os campos utilizados pelo formato ID3v1.

Campo	Comprimento (bytes)	Descrição
Cabeçalho	3	Valor de verificação “TAG”
Título	30	Título da música representado com caracteres ASCII
Artista	30	Nome do artista representado com caracteres ASCII
Álbum	30	Nome do álbum representado com caracteres ASCII
Ano	4	Ano representado por 4 dígitos (caracteres ASCII e não um inteiro)
Comentário	28 ou 30	Comentário associado à música
Byte-zero	1	Se o número da faixa estiver incluído então deverá ser 0. Nesse caso apenas são usados 28 bytes no comentário
Número	1	Número da faixa no álbum representado por um inteiro
Género	1	Género da música representado por um inteiro e definido de acordo com a tabela disponibilizada em: http://www.id3.org/id3v2.3.0#head-129376727e5309c1de1888987d070288d7c7e7

Utilize o ficheiro **musica.mp3** para testar o programa que desenvolveu.

Exemplo usando o ficheiro musica.mp3

Título - Particule
Artista - Silence
Album - L'autre endroit
Ano - 2006
Número - 7
Género - Instrumental

3 – As páginas *web* utilizam HTML como linguagem de anotação. Os conteúdos podem ser associados a *tags*, conferindo-lhes diferentes propriedades. Uma dessas *tags* é a de ligação a outras páginas que é definida, por exemplo, assim:

`Página da FEUP`

Escreva um programa que leia um ficheiro HTML e liste todas as ligações para outras páginas e nomes associados. A linguagem HTML é flexível na colocação de espaços e aspas nas *tags* mas neste exercício considere que todas as ligações são definidas exatamente como no exemplo anterior.

Utilize o ficheiro **pagina.html** para testar o programa que desenvolveu.

Exemplo usando o ficheiro pagina.html

Página da FEUP - <http://fe.up.pt>
Página de Prog2 no SiFEUP - [https://www.fe.up.pt/si/\(...\)EEC0009&p_periodes=2S](https://www.fe.up.pt/si/(...)EEC0009&p_periodes=2S)
Página da Prog2 no Moodle - <https://moodle.fe.up.pt/1011/course/view.php?id=797>
Livro "C How to Program" - <http://proquestcombo.safaribooksonline.com/9780136085881>

4 – O ficheiros BMP (Bitmap) são um formato muito popular para armazenamento de imagens que surgiu com os Sistemas Operativos OS/2 e Windows 3.0. Escreva um programa que leia informação de um ficheiro BMP, tendo em conta a especificação do formato indicado nas seguintes tabelas.

De uma forma simplificada, o ficheiro BMP tem a seguinte estrutura:

Campo	Comprimento (bytes)	Descrição
Cabeçalho Bitmap	14	Armazena informação geral do ficheiro Bitmap.
Cabeçalho DIB	108 (na versão 4)	Armazena informação detalhada sobre a imagem e define o formato de armazenamento de cada pixel
Array de pixéis	Variável	Valores dos pixéis da imagem. A cada linha são acrescentados bytes a 0 se o tamanho total dessa linha em bytes não for múltiplo de 4 (não é o caso da imagem de teste)

O cabeçalho Bitmap tem a seguinte estrutura:

Comprimento (bytes)	Descrição
2	Indicação de início do cabeçalho. Para BMP o valor é 0x42 0x4D, ou seja “BM” em ASCII
4	Tamanho total do ficheiro BMP em bytes
2	reservado
2	reservado
4	Offset, ou seja, endereço do byte em que o array de pixéis começa

O cabeçalho DIB na versão 4 tem a seguinte estrutura:

Comprimento (bytes)	Descrição
4	Tamanho deste cabeçalho (108 bytes)
4	Largura da imagem em pixéis (inteiro com sinal)
4	Altura da imagem em pixéis (inteiro com sinal)
2	Número de planos de cor usados; deve ser 1
2	Número de bits por pixel; valores típicos: 1, 4, 8, 16, 24 e 32
4	Método de compressão usado
4	Tamanho da imagem em bytes, ou seja, o tamanho do <i>array</i> de pixéis e não do ficheiro
...	Resto do cabeçalho, que não é relevante para este exercício

No *array* de pixels a imagem é guardada linha a linha, começando pela última linha até à primeira. No caso de imagens de 24 bits, como a imagem de teste, cada pixel é definido por 3 bytes que correspondem aos valores de B, G e R, respetivamente.

Mais informações sobre o formato BMP podem ser encontradas na respetiva página na Wikipedia: http://en.wikipedia.org/wiki/BMP_file_format

Utilize o ficheiro **imagem.bmp** para testar o programa que desenvolveu.

*Exemplo usando o ficheiro **imagem.bmp***

```
Tamanho total do ficheiro BMP: 1279322
Largura da imagem: 800
Altura da imagem: 533
Bits por pixel: 24
Valor RGB no pixel (0,0): 54 59 62
Valor RGB no pixel (532,0): 70 97 108
Valor RGB no pixel (408,443): 179 91 71
```

Exercício complementar

5 – Escreva um programa que analise os seus ficheiros de código fonte em C e apresente as seguintes estatísticas para um determinado ficheiro:

- Número de linhas de código
- Tamanho médio de cada linha de código
- Número de caracteres
- Número de vezes que utiliza as funções da biblioteca *standard C* `printf` e `scanf`.
- Número de comentários (começados com `/*` e terminados com `*/`)

Utilize os ficheiros que criou nesta ficha de exercícios para testar o seu programa.