Programação

Licenciatura em Engenharia Informática: 1º ano - 2º semestre 2019/2020

Guião laboratorial n.º 3

Ficheiros

Tópicos da matéria:

- Ficheiros de texto e ficheiros binários
- Operações com ficheiros

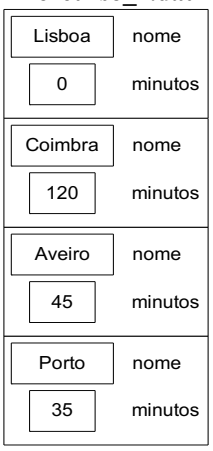
Bibliografia:

K. N. King, *C Programming: A Modern Approach*: Capítulo 22.

A – Ficheiros Binários: Operações Simples

1. Assuma que se pretende gerir percursos efetuados pelos comboios da CP, sabendo que cada um desses percursos é armazenado num ficheiro binário individual. O que caracteriza cada um dos percursos são os seus pontos de paragem. Assim, num ficheiro binário relativo a um determinado percurso estão armazenados os vários pontos de paragem e o tempo que demora a efetuar cada uma das ligações entre duas paragens consecutivas. A informação sobre uma paragem é armazenada numa estrutura do tipo `struct paragem`.

"Percurso_1.dat"




```
struct paragem{
    char nome[50];
    int minutos;
};
```

O campo `nome` armazena o nome da paragem e o campo `minutos` armazena o tempo que o comboio demora (em média) desde a paragem anterior. As estruturas no ficheiro estão armazenadas por ordem de passagem. A primeira estrutura corresponde ao local de partida do comboio e a última ao local de destino. Na estrutura relativa ao ponto de partida do comboio o campo `minutos` tem o valor 0. Se existir um percurso "Lisboa-Coimbra-Aveiro-Porto" o ficheiro binário correspondente poderia ficar com o formato indicado na figura.

a) Escreva uma função que mostre no monitor os nomes e os minutos de todas as estruturas que fazem parte do ficheiro com estas propriedades. O nome do ficheiro é passado como parâmetro da função.

b) Escreva uma função que calcule a duração total em minutos do percurso completo armazenado num ficheiro. A função recebe o nome do ficheiro como parâmetro e devolve a duração contabilizada.

c) Desenvolva uma função que verifique se um determinado percurso de um comboio permite efetuar a ligação entre dois locais A e B. A função recebe, como parâmetros, o nome do ficheiro binário onde está armazenado o percurso e os nomes dos locais de início (A) e de final (B) da ligação pretendida. Caso a ligação possa ser efetuada pelo percurso armazenado no ficheiro, a função deve devolver o número de minutos que esta demora. Se a ligação não puder ser efetuada, a função deve devolver -1.

Nota: tenha em atenção que os locais A e B não têm que ser adjacentes no percurso do comboio. A única restrição é a de que o percurso do comboio deve passar primeiro em A e depois em B. Por exemplo, considerando o percurso armazenado no ficheiro do exemplo, as ligações Coimbra - Porto ou Lisboa - Aveiro são válidas, enquanto que a ligação Aveiro - Coimbra é inválida.

2. Um ficheiro binário tem vários valores inteiros armazenados.

a) Escreva uma função determine a soma e a média de todos os valores armazenados. A função recebe como parâmetro o nome do ficheiro binário. Devolve a soma e a média como resultado.

b) Escreva uma função que crie um vetor dinâmico para onde deve copiar todos os **valores pares** que existem no ficheiro binário. A função tem o seguinte protótipo:

```
int* criaVetor(char *nomeFich, int *tam);
```

A função devolve o endereço do vetor dinâmico como resultado. Recebe como parâmetros o nome do ficheiro binário e o endereço de uma variável inteira onde deve colocar a dimensão do vetor. Se ocorrer algum problema (ficheiro não existente, erro de alocação ou ficheiro sem valores pares), a função devolve NULL e coloca o valor 0 na variável referenciada pelo segundo parâmetro.

c) Escreva uma função que crie 2 novos ficheiros binários com informação obtida do ficheiro original: num deles devem ficar os valores superiores ou iguais à média e no outro os valores inferiores à média. O ficheiro original não deve ser alterado. A função recebe os nomes dos 3 ficheiros como parâmetro.

3. Considere a seguinte definição:

```
struct cliente{
      char nome[200];
      char morada[200];
      int conta;
      int montante;
};
```

a) Um banco tem informação armazenada sobre os seus clientes num ficheiro binário contendo estruturas do tipo `struct cliente`. Escreva uma função que mostre no monitor os nomes e os números de conta de todos os clientes que fazem parte do ficheiro. O nome do ficheiro é passado como parâmetro da função.

b) Escreva uma função que mostre no monitor o nome e o número de conta do cliente com o saldo mais elevado. O nome do ficheiro binário onde estão guardados os clientes é passado como parâmetro da função.

B – Ficheiros de Texto: Operações Simples

4. Escreva uma função que mostre o conteúdo de um ficheiro de texto na consola. O nome do ficheiro é passado como parâmetro.

5. Escreva uma função que duplique um ficheiro de texto. Os nomes dos ficheiros original e da cópia devem ser passados como parâmetro.

6. Altere a função da questão anterior de modo a garantir que no ficheiro cópia as linhas do texto surjam numeradas. O número máximo de caracteres numa linha não é conhecido à partida.

Ficheiro de Texto Original

```
aaa
bbb bbbb
c
ddddd
```

Ficheiro de Texto Cópia

```
1. aaa
2. bbb bbbb
3. c
4. ddddd
```

7. Escreva uma função que mostre no monitor uma determinada linha de um ficheiro de texto. O nome do ficheiro e o número da linha são passados como parâmetros da função.

8. Escreva uma função que descubra qual a vogal mais comum num ficheiro de texto. A função recebe o nome do ficheiro como parâmetro e devolve a vogal mais comum.

9. Num ficheiro de texto estão armazenadas as notas parciais (em percentagem) de um conjunto de alunos. Cada aluno efetuou alguns testes com um determinado peso (o número máximo de testes é 6). Pretende-se calcular a nota final para cada um dos alunos que fazem parte do ficheiro. A estrutura genérica do ficheiro inicial é a seguinte:

Numero de alunos: 34	linha onde é indicado o número de alunos
Numero de provas: 4	linha onde é indicado o número de provas
Peso das provas: 15 15 30 40	linha com o peso de cada uma das provas
	linha em branco
Aluno 1: 90 75 50 70	a partir deste ponto existe uma linha para cada
Aluno 2: 23 34 45 60	aluno com as suas notas parciais
…	
Aluno 34: 8 67 42 65	

Escreva uma função que receba o ficheiro original e crie um novo ficheiro de texto contendo as notas finais dos alunos (em percentagem). A estrutura do ficheiro a criar é ilustrada à direita: Na primeira linha deve surgir a indicação do número de alunos, seguida pelas notas finais calculadas para cada aluno do ficheiro original. A função recebe os nomes dos 2 ficheiros como parâmetros.

```
Numero de alunos: 34
Aluno 1: 68%
Aluno 2: 46%
...
Aluno 34: 50%
```

C – Outras Operações com Ficheiros

10. Considere a seguinte definição:

```
struct cliente{
      char nome[200];
      char morada[200];
      int conta;
      int montante;
};
```

Um banco tem informação armazenada sobre os seus clientes num ficheiro binário contendo estruturas do tipo `struct cliente`. O ficheiro está **ordenado alfabeticamente** pelo nome do cliente. Pode assumir que não existem clientes diferentes com nomes iguais.

a) Escreva uma função que mostre no monitor os nomes e os números de conta de todos os clientes que fazem parte do ficheiro. O nome do ficheiro é passado como parâmetro da função.

b) Escreva uma função que mostre no monitor o número de bytes do ficheiro e o número de clientes que aí estão armazenados. O nome do ficheiro é passado como parâmetro da função.

c) Escreva uma função que mostre no monitor os nomes e os números de conta de todos os clientes que fazem parte do ficheiro. A informação deve ser listada por ordem alfabética inversa. O nome do ficheiro é passado como parâmetro da função.

d) Escreva uma função que efetue uma transferência entre 2 clientes. A função tem o seguinte protótipo:

```
int transfere(char *nomeF, char *or, char *dest, int valor);
```

Transfere *valor* da conta do cliente *or* para o cliente *dest*. Ambos os clientes devem fazer parte do ficheiro *nomeF*. A função deve atualizar os montantes dos clientes, de acordo com o valor transferido. A transferência só é concretizada se existir saldo suficiente na conta do cliente *or*. Devolve 1 se tudo correu bem, ou 0, caso contrário.

e) Escreva uma função que elimine um cliente do ficheiro. A função recebe como parâmetros o nome do ficheiro e o nome do cliente a eliminar.

f) Escreva uma função que adicione um novo cliente ao ficheiro de clientes do banco. A função recebe como parâmetros o nome do ficheiro e a estrutura com a informação sobre o novo cliente (os campos da estrutura já estão preenchidos). A função deve garantir que a estrutura com informação do novo cliente é inserida no ficheiro na posição correta, de modo a que este continue ordenado.

g) Considere que o banco dividiu por dois ficheiros a informação sobre os seus clientes. Em cada um deles as estruturas estão armazenadas por ordem alfabética do nome do cliente (recorde que não existem clientes diferentes com nomes iguais). O banco pretende reunir as estruturas num só ficheiro. Escreva uma função para realizar esta tarefa. Os nomes dos dois ficheiros originais e do novo ficheiro são passados como parâmetros. No novo ficheiro, as estruturas devem continuar por ordem alfabética. Podem existir duplicações da informação de um cliente (ou seja, o mesmo cliente tem uma estrutura em cada um dos ficheiros originais). Neste caso, só uma estrutura deve ser escrita no novo ficheiro e o valor do montante deve ser a soma dos dois montantes existentes em cada uma das cópias. Pode assumir que nestes casos a morada e o número de conta são iguais.

11. As temperaturas mínima e máxima verificadas em Coimbra durante cada um dos dias de um ano comum estão guardadas num ficheiro. Cada uma destas temperaturas foi armazenada como sendo um número real de precisão simples (tipo *float*). O modo como os valores estão armazenados é o seguinte:

```
Mínima do dia 1 de Janeiro
Máxima do dia 1 de Janeiro
Mínima do dia 2 de Janeiro
Máxima do dia 2 de Janeiro
…
Mínima do dia 31 de Dezembro
Máxima do dia 31 de Dezembro
```

a) Considerando que os dados estão armazenados num ficheiro de texto:

i) Desenvolva um programa que indique quais as temperaturas mínima e máxima do dia 4 de Fevereiro

ii) Desenvolva um programa que indique a média das temperaturas mínimas e máxima para cada um dos meses do ano.

b) Considerando que os dados estão armazenados num ficheiro binário:

i) Desenvolva um programa que indique quais as temperaturas mínima e máxima do dia 4 de Fevereiro

ii) Desenvolva um programa que indique a média das temperaturas mínimas e máxima para cada um dos meses do ano.

12. Num ficheiro de texto encontram-se armazenadas todas as apostas do totoloto de uma determinada semana. Em linhas consecutivas surge o nome do apostador e na seguinte os seis números em que apostou (os números estão armazenados por ordem crescente). A informação está separada por um ou mais espaços e não existem linhas em branco entre a informação referente aos diferentes apostadores.

Extrato do ficheiro de texto:

```
…
Joao C. Silva
2     3     6     8     10    34
Ana Lima Nunes
4     10    23    30    31    38
Antunes Silvestre
3     12    40    41    42    43
…
```

Desenvolva um programa que elimine do ficheiro todos os apostadores que não ganharam nenhum prémio. Considere que um apostador ganha um prémio se acertou pelo menos em três dos números sorteados. O programa recebe o nome do ficheiro e os seis números que foram sorteados através da linha de comando. Os números sorteados são especificados na linha de comando por ordem crescente.

13. O Instituto de Meteorologia armazena num ficheiro de texto os dados mensais relativos às temperaturas e horas de sol verificadas na região de Coimbra. Tomando como exemplo o mês de Agosto, a estrutura do ficheiro é a seguinte:

```
Precipitação no mês de Agosto na região de Coimbra
Técnico responsável: Artur Almeida Santos

Dia  Mínima    Máxima    Nascer_sol     Por_Sol   Horas_Sol
1    12        28        6:45      20:45          14:00
2    14        33        6:45      20:40          13:55
3    15        33        6:50      20:38          13:48
…
31   13        26        7:01      20:22          13:21
```

O ficheiro tem duas linhas de cabeçalho com informação genérica. Após essas linhas, tem 6 colunas com informação sobre temperaturas e número de horas de sol em cada um dos dias do mês. A informação relativa a cada dia do mês está armazenada numa mesma linha, separada por um ou mais espaços em branco.

a) Pretende-se que escreva um programa que funcione como um filtro da informação existente no ficheiro. O objetivo é criar um novo ficheiro de texto para o qual é copiada a informação de apenas duas colunas: a coluna que indica qual o dia do mês e uma outra escolhida pelo utilizador. Cada uma das colunas tem associado um número inteiro (coluna `Dia` é a 1, coluna `Mínima` é a 2, coluna `Máxima` é a 3, e assim sucessivamente). Se, por exemplo, for selecionada a coluna 6 para filtragem, o novo ficheiro deverá ficar com a seguinte estrutura:

```
Precipitação no mês de Agosto na região de Coimbra
Técnico responsável: Artur Almeida Santos

Dia  Horas_Sol
1    14:00
2    13:55
3    13:48
…
31   13:21
```

O nomes dos ficheiros (original e final) e o número da coluna a filtrar são passados como argumento da linha de comando.

b) Pretende-se armazenar a informação original num ficheiro binário. Sugira uma arrumação possível para essa informação.

14. Pretende-se preencher algumas posições de uma matriz de números reais de precisão simples (tipo `float`). Os valores a utilizar para o preenchimento estão armazenados num ficheiro binário. A informação no ficheiro está armazenada no seguinte formato: sequências de dois números inteiros X, Y seguidos de um número real Z. Esta informação significa que o valor real Z deve ser utilizado para preencher o elemento da matriz na intersecção da linha X com a coluna Y. No ficheiro binário existem sucessivas sequências destes conjuntos de 3 números (tantas quantas as posições que devem ser inicializadas). Na figura pode ver um exemplo de um possível ficheiro binário com duas sequências de inicialização:

3 4 12.1 0 1 -2.3

Considerando este exemplo, deveriam ser preenchidas as duas posições da matriz:

- atribuir ao elemento na linha 3, coluna 4 o valor 12.1
- atribuir ao elemento na linha 0, coluna 1 o valor -2.3

Escreva uma função que efetue o preenchimento da tabela. A função recebe, como argumentos, informação completa sobre a matriz (o endereço do primeiro elemento da matriz, o número de linhas e o número de colunas) e o nome do ficheiro binário. No final, a função deve devolver o número de posições da tabela que foram preenchidas com valores lidos do ficheiro. Todas as posições da matriz que não forem preenchidas com valores lidos do ficheiro devem ser inicializadas com o valor 0.

Nota: A função deve verificar se as posições especificadas no ficheiro para preenchimento são válidas (i.e., se não estão fora dos limites da matriz).

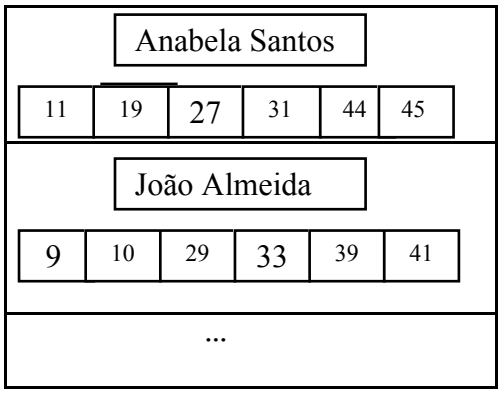
15. Considere que num ficheiro de texto existe informação sobre a chave do totoloto do concurso referente a uma determinada semana. A informação é constituída pelo número e data do concurso e pelos números sorteados. Esta informação está dividida por várias linhas de acordo com a estrutura exemplificada na figura seguinte:

"texto.txt"

Concurso nº: 427 Sorteio efetuado em: 02/09/00 Chave: 5 12 17 29 40 43 Suplementar: 14

Num ficheiro binário está por sua vez armazenada informação sobre todas as apostas efetuadas nessa semana. A informação individual de cada apostador é armazenada numa estrutura do tipo `struct aposta`. Na figura seguinte pode ver-se um exemplo parcial da arrumação das estruturas relativas a alguns apostadores num possível ficheiro.

"bin.dat"




```
struct aposta{
      char nome[50];
      int nums[6];
};
```

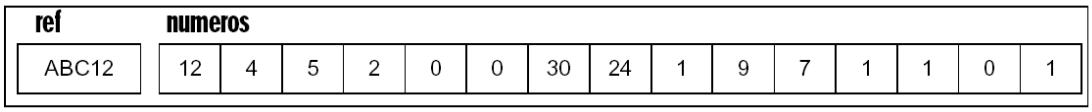
Em cada uma das estruturas o campo `nome` armazena o nome do apostador e a tabela `nums` armazena os números da aposta, ordenados de forma crescente.

Escreva um programa que elimine do ficheiro binário os dados das pessoas que não acertaram em pelo menos três dos números sorteados. O programa recebe os nomes dos ficheiros (ficheiro de texto com informação sobre o sorteio e ficheiro binário com informação sobre os apostadores) como argumentos da linha de comando.

16. A sapataria Moderna informatizou a gestão do stock dos sapatos que existem na sua loja. Para cada modelo existente, a informação a guardar é a sua referência e a quantidade de sapatos em stock para cada um dos tamanhos possíveis. Esta informação pode ser armazenada numa estrutura do tipo:

```
struct sapato{
      char ref[11];
      int numeros[15];
};
```

O campo ref armazena a referência alfanumérica que identifica o sapato. O campo `numeros` armazena a quantidade de sapatos com esta referência que existem em stock, para cada um dos tamanhos possíveis. A identificação de quantos sapatos existem para cada número é feita da seguinte forma: os tamanhos, para todos os modelos, variam entre 31 e 45. Cada posição da tabela `numeros` armazena o número de pares existentes para um tamanho particular. Na figura pode-se ver um exemplo de uma estrutura relativa ao modelo com referência ABC12. Neste caso, existem 12 pares de sapatos número 31, 4 pares número 32, 5 pares número 33, e assim sucessivamente.



Implemente uma função que transfira a informação do ficheiro binário para um ficheiro de texto. No ficheiro de texto, a informação deve ficar organizada da seguinte forma: para cada tamanho possível, indicação do número de sapatos em stock para cada um dos modelos comercializados pela sapataria. Um exemplo desta organização pode ser conferido a seguir:

Ficheiro de texto:

```
Tamanho 31:
Ref ABC12: 12 pares
Ref XX456: 6 pares
Ref 890AA: 0 pares
Ref ASWQ: 2 pares

Tamanho 32:
Ref ABC12: 4 pares
Ref XX456: 2 pares

....
```

A função recebe como parâmetros os nomes dos ficheiros a manipular.

17 . Uma loja de material de escritório informatizou o processo de cálculo das vendas anuais de todos os seus vendedores. Para cada vendedor, a loja possui o total de vendas (em euros) que este realizou em cada mês do ano. Esta informação pode ser armazenada numa estrutura do tipo:

```
struct vendas{
   char nome[50];
   float totais[12];
};
```

O campo `nome` armazena o nome do vendedor e o campo `totais` armazena as vendas que este efetuou nos 12 meses do ano.
Suponha que a informação relativa ao ano de 2004 foi já armazenada num **ficheiro binário**. Neste ficheiro encontram-se gravadas um conjunto de estruturas do tipo acima definido, já devidamente preenchidas com os nomes e respetivos totais realizados por mês.
Desenvolva uma função em C que crie um **ficheiro de texto**, a partir da informação contida no ficheiro binário No ficheiro de texto, a informação deve ficar organizada da seguinte forma: para cada mês do ano, qual foi o total de vendas realizadas por todos os vendedores. A última linha do ficheiro deve conter qual o mês de maior lucro e respetiva faturação. Um exemplo desta organização pode ser conferido a seguir:

```
MES    TOTAL DE VENDAS
1      2480.00
2      3355.00
3      7163.00
4      3246.00
5      2482.00
...
11     1707.00
12     2402.00
Melhor mês do ano: 3 ----> 7163.00
```

A função recebe como argumentos os nomes dos dois ficheiros a manipular.