

Universidade da Beira Interior Departamento de Informática

1. O clube “Chá das 5” é uma instituição sem fins lucrativos que organiza encontros nas casas dos seus membros, com vista à reflexão sobre temas importantes nas sociedades actuais. O clube mantém uma lista dos seus membros, contendo informação sobre o BI, nome e data de admissão. Para cada encontro, será importante registar a data, o sócio que o organizou (apenas 1), o tema do encontro e informação sobre os sócios que estiveram presentes.

1a) (2 valores) Defina as estruturas de dados que achar convenientes para uma solução computacional que guarde a informação pretendida para o clube “Chá das 5”.

1b) (2 valores) Mediante as estruturas por si definidas na alínea anterior, esquematize a organização da informação resultante do armazenamento dos seguintes itens:

- Membro 123456, José Andrade, 1/10/2000
- Membro 321321, Vitor Coelho, 2/11/2002
- Membro 123431, Andreia Soares, 3/11/2006
- Membro 163435, Rita Ferreira, 13/12/2009

- Encontro 12/10/2014, Organização 123456, “O buraco do ozono”, Presentes: 123456, 321321, 123431
- Encontro 22/11/2014, Organização 321321, “Espécies animais ameaçadas”, Presentes: 321321, 163435
- Encontro 22/11/2014, Organização 321321, “Turismo na Beira Interior”, Presentes: 321321

1c) (3 valores) Implemente uma função que devolva o número médio de elementos presentes nos encontros organizados por um determinado membro. (adapte o protótipo da função às estruturas de dados por si definidas).

Protótipo: float mediaPresentes(..., int BIOrganizador);

1c) (3 valores) Codifique a função que lista o nome dos membros que nunca organizaram nenhum encontro.

Protótipo: void mostraInactivos(...);

1d) (2 valores) Implemente uma função que devolva o total de encontros realizados num determinado ano civil.

Protótipo: int totalAno(..., int ano);

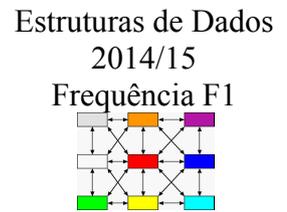
2) Considere a seguinte estrutura de dados referente a uma lista de salto:

```
typedef struct NODO{
    int ID;           //Chave de ordenação na lista
    int nivel;       //Nível do elemento
    struct NODO *nseg[MAX_NIVEIS];
}Nodo;
```

2a) (2 valores) Implemente uma função que verifique se a lista está bem ordenada (crescentemente). A função deverá devolver 1 em caso afirmativo, ou 0 em caso contrário:

Protótipo: int bemOrdenada(Nodo *L);

2b) (3 valores) Implemente uma função que verifique se a lista está bem formada, segundo o conceito



tipicamente associado a listas de salto e uma determinada proporção $p \in]1, \infty[$ Isto é:

total nós nível 0 $> p * (\text{total nós nível } 1)$

total nós nível 1 $> p * (\text{total nós nível } 2)$

...

total nós nível (MAX_NIVEL-2) $> p * (\text{total nós nível (MAX_NIVEL-1)})$.

A função deverá devolver 1 em caso afirmativo, ou 0 caso contrário.

Protótipo: `int bemFormada(Nodo *L, float p);`

2c) (3 valores) Implemente uma função que mostre o ID do elemento a pesquisar para o qual a utilização do mecanismo de procura em listas de salto se revelaria mais vantajoso em relação à procura puramente sequencial, isto é, em que a diferença de nós visitados nas 2 formas de pesquisa seja máxima.

Protótipo: `void IDMaisVantajoso(Nodo *L);`