UNIVERSIDADE DA BEIRA INTERIOR

Algoritmos e Estruturas de Dados

Turno: EXTRA

Mini-teste prático 1 (2.0 val)

30 min

20/03/2024

Nº _____ Nome _____

(preencher o cabeçalho com número e nome, e entregar no fim do teste)

LISTAS LIGADAS SIMPLES

Considere as seguintes definições dos tipos de dados **INFO**, **Nodo** e **PNodo** (apontamentos das aulas):

```
typedef struct {
    int  numAluno;    // chave (valor único), que é um número positivo (> 0)
    float notasMTP;  // soma das notas dos mini-testes práticos ([0..4])
    float notasTE;   // soma das notas dos testes escritos/frequências ([0..16])
    int  notaFinal;  // valor arredondado da nota final (notasMTP + notasTE)
} INFO;

struct Nodo {
    INFO Elemento;
    struct Nodo *Prox;
};

typedef struct Nodo *PNodo;
```

Cada elemento (registo) do tipo **INFO** corresponde às notas obtidas por um estudante na disciplina de AED, no qual os valores do campo **numAluno** devem ser únicos (chave).

Usando as definições apresentadas e as funções já implementadas necessárias e contidas nas bibliotecas dadas, resolva os seguintes problemas:

- [1.2 val]** Implementar uma função em C que,
 - **receba** uma lista **L** com elementos do tipo **INFO** e um valor inteiro **nota**,
 - **devolva** o **menor** valor do campo **numAluno** entre todos os elementos de **L** com valor no campo **notaFinal** igual a **nota**, caso exista (se não existir, deve devolver o valor **negativo** qualquer)
- [0.8 val]** Implementar um programa em C que realize as seguintes ações (pela ordem indicada):
 - a) construa uma lista com elementos do tipo **INFO**, **Lista**, cujo tamanho é um valor entre **1** e **15** => usar a função já implementada **criarListaAleatoria**
 - b) mostre no monitor todos os elementos da lista **Lista** => usar uma **função adequada** de uma das bibliotecas dadas
 - c) mostre no monitor os valores dos campos **notasMTP** e **notasTE** do elemento que se encontra na segunda posição da lista **Lista** (o primeiro elemento é a cabeça da lista), caso exista
 - d) atribua à variável **NOTA** o valor do campo **notaFinal** do primeiro elemento (cabeça) da lista **Lista** e mostre o valor da variável **NOTA** no monitor
 - e) determine e mostre o **menor** valor do campo **numAluno** entre todos os elementos da lista **Lista** com valor no campo **notaFinal** igual a **NOTA** => **usar a função implementada em 1.**