

Listas ligadas com ligações simples

A. Considere as seguintes definições dos tipos de dados **INFOLista**, **NodoLista** e **PNodoLista** (apontamentos das aulas teóricas):

```
typedef struct {  
    int numAluno;           // número de aluno ({ 70000, ..., 75000 })  
    float notasMTP[2];      // notas dos mini-testes práticos ([0.0, 2.0])  
    float notasTE[2];       // notas dos testes escritos ([0.0, 8.0])  
    int notaFinal;          // nota final ({ 0, ..., 20 }): valor arredondado da soma das notas  
} INFOLista;  
  
struct NodoLista {  
    INFOLista Elemento;  
    struct NodoLista *Prox;  
};  
  
typedef struct NodoLista *PNodoLista;
```

Cada elemento (registo) do tipo de dados **INFOLista** corresponde aos dados sobre a avaliação de um aluno à disciplina de AED, durante a época de Aprendizagem. O campo **numAluno** é a chave (valor único para cada elemento do tipo **INFOLista**).

Descarregar as seguintes bibliotecas (**Folhas práticas ---> Bibliotecas e Exercícios das folhas práticas ---> Listas ligadas simples**):

Aleatorio.h ---> operações para gerar números aleatoriamente
EADLista.h ---> operações básicas sobre a EAD **Lista**
OperacoesBasicasExA.h ---> operações básicas sobre o tipo de dados **INFOLista**
OperacoesExA.h ---> operações a implementar relativas aos exercícios do **grupo A**
MainExA.c ---> programa principal relativo aos exercícios do **grupo A**

Elaborar um programa em C que utilize as operações contidas nas bibliotecas referidas e resolva as questões colocadas a seguir, acrescentado-as uma a uma ao programa.

1. Implementar uma função que,
 - receba dois valores inteiros positivos, **A** e **B**,
 - crie e devolva uma lista com elementos do tipo **INFOLista** gerados aleatoriamente (usar o gerador de números - biblioteca **Aleatorio**), cujo tamanho é um valor entre A e B.Usar esta função para construir uma lista de elementos do tipo **INFOLista**, cujo tamanho é um valor gerado aleatoriamente entre 0 e 15.
2. Mostrar a lista criada em 1, em que os elementos são apresentados
 - do início para o fim, e
 - do fim para o início.

3. Implementar uma função iterativa que
 - receba uma lista **L** com elementos do tipo **INFOLista**,
 - devolva o **tamanho** (número de elementos) de **L**.
4. Implementar uma função recursiva que
 - receba uma lista **L** com elementos do tipo **INFOLista** e um valor inteiro **N**,
 - devolva a quantidade de elementos de **L** com valor no campo **notaFinal** igual a **N**.
5. Implementar uma função iterativa que
 - receba uma lista **L** com elementos do tipo **INFOLista** e um valor inteiro **N**,
 - devolva a quantidade de elementos de **L** com valor no campo **notaFinal** maior ou igual a **N**.
6. Implementar uma função (iterativa ou recursiva) que
 - receba uma lista **L** com elementos do tipo **INFOLista**,
 - devolva o **maior** valor no campo **notaFinal** de **L**.
7. Implementar uma função (iterativa ou recursiva) que
 - receba uma lista **L** com elementos do tipo **INFOLista**,
 - devolva o aluno com melhor nota nas frequências (o valor do campo **numAluno** com o maior valor na soma dos valores dos campos **notasTE** de **L**.
8. Implementar uma função (iterativa ou recursiva) que
 - receba uma lista **L** com elementos do tipo **INFOLista**,
 - devolva o aluno com melhor nota em F1 + F2 (o valor do campo **numAluno** do elemento ed L com o maior valor na soma dos campos **notasTE**).
9. Implementar uma função (iterativa ou recursiva) que
 - receba uma lista **L** com elementos do tipo **INFOLista**,
 - devolva o aluno com pior nota nos mini-testes práticos (o valor do campo **numAluno** do elemento de L com o menor valor num dos campos **notasMTE**).
10. Implementar uma função iterativa que
 - receba uma lista **L** com elementos do tipo **INFOLista**,
 - devolva o valor **médio** dos valores do campo **notaFinal** de L.
11. Implementar uma função (iterativa ou recursiva) que
 - receba uma lista **L** com elementos do tipo **INFOLista** e um inteiro **K**,
 - devolva o menor elemento de **L** com valor no campo **notaFinal** maior ou igual a **K**.
12. Implementar uma função que
 - receba uma lista **L** com elementos do tipo **INFOLista** e dois valores inteiros, **K1** e **K2**,
 - devolva o número de elementos de **L** com valores do campo **notaFinal** entre **K1** e **K2**.
13. Implementar uma função que
 - receba uma lista **L** com elementos do tipo **INFOLista** e um inteiro **num**,
 - remova o elemento da lista **L** com valor no campo **numAluno** igual a **num**.

14. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista** e um inteiro **K**,
- remova todos os elementos de **L** cujo valor no campo **notaFinal** seja **menor** do que **K**.

15. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista** e um inteiro **nota**,
- remova todos os elementos da lista **L** com valor no campo **notaFinal** igual a **nota**.

16. Implementar uma função que,

- receba uma lista **L** com elementos do tipo **INFOLista** e um número inteiro positivo **N**,
- remova de **L** os seus **N** primeiros elementos (caso **N** seja maior do que o comprimento da lista, todos os seus elementos devem ser removidos).

17. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista** e um número inteiro positivo **N**, e
- divida a lista em duas, de tal forma que a segunda lista comece com o elemento que se segue ao primeiro elemento com valor no campo **numAluno** igual a **num**.

18. Implementar um função que

- receba uma lista **L** com elementos do tipo **INFOLista**, e
- ordene-a por ordem crescente do campo **notaFinal**.

19. Implementar uma função que,

- receba uma lista **L** com elementos do tipo **INFOLista** ordenados crescentemente pelo campo **notaFinal** e um valor inteiro **num**,
- remova todos os elementos de **L** com valores no campo **notaFinal** iguais a **Num**, caso existam.

20. Implementar uma função que,

- receba uma lista **L** com elementos do tipo **INFOLista** ordenados crescentemente pelo campo **notaFinal** e um inteiro **num**,
- remova todos os elementos de **L** com valores no campo **notaFinal** menores ou iguais a **num**, caso existam.

21. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista** e dois valores inteiros, **K1** e **K2**,
- devolva o número de elementos de **L** com valores do campo **numFinal** maiores que **K1** e menores que **K2**.

22. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista**,
- devolva o elemento de **L** com o segundo maior valor do campo **notaFnal** (percorrer a lista apenas uma vez).

23. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista**,
- remova o primeiro elemento e o último elemento da lista **L**.

24. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista**,
- remova todos os elementos de **L** com exceção do primeiro elemento e do último elemento.

25. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista**,
- passe o último elemento para a segunda posição da lista **L**, mantendo a ordem dos restantes elementos.

26. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista**,
- inverta a lista **L** (o último elemento passa a primeiro, o penúltimo a segundo, e assim sucessivamente).

27. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista** e um número inteiro **N**,
- remova o **N-ésimo** elemento da lista **L**.

B. Considere a seguinte definição do tipo de dados **INFOLista**:

```
typedef struct{
    int numCC;    // chave e com valores entre 100000 e 200000
    int anoNasc;  // com ano entre 1920 e 2024
    float altura; // em metros (ex: 1.86)
    int genero;   // 1 = feminino e 2 = masculino
} INFOLista;
```

Elaborar um programa que utilize as operações que se encontram nas bibliotecas criadas antes e resolva as questões colocadas a seguir, uma a uma.

1. Implementar uma função que

- receba dois valores inteiros positivos, **A** e **B**,
- crie e devolva uma lista com elementos do tipo **INFOLista** gerados aleatoriamente (usar o gerador de números - biblioteca **Aleatorio**), cujo tamanho é um valor entre A e B.

Usar esta função para construir uma lista de elementos do tipo **INFOLista**, cujo tamanho é um valor gerado aleatoriamente entre 0 e 15.

2. Mostrar a lista criada antes, em que os elementos são apresentados

- do início para o fim, e
- do fim para o início.

3. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista** e um número real **alt** ($1.0 \leq \text{alt} \leq 2.2$)
- devolva a quantidade de elementos da lista **L**, cujo valor do campo **altura** é maior que **alt**

4. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista** e um número inteiro **A** ($1920 \leq A \leq 2024$)
- devolva a quantidade de elementos de L nascidos no ano **A**.

5. Implementar uma função que,

- receba uma lista **L** com elementos do tipo **INFOLista** e um inteiro positivo **N** ($1 \leq N \leq 10$),
- remova da lista **L** os seus **N** primeiros elementos; se **N** é maior que o comprimento da lista, então todos os seus elementos devem ser removidos, ficando a lista L vazia.

6. Implementar uma função que

- receba uma lista **L** com elementos do tipo **INFOLista**,
- devolva 2 listas do mesmo tipo (DADOS), uma com todos os elementos do género feminino (campo **genero** = 1) e a outra com todos os elementos do género masculino (campo **genero** = 2).