

Lista 10**Variações de listas; Tipo Abstrato de Dados (TAD)****Questão 1**

Faça uma função que busca um elemento **x** em uma **lista ligada circular** (**No *L**), devolvendo o ponteiro para o **primeiro** nó encontrado ou **NULL** se o elemento não estiver na lista.

```
1 No* busca_lista(No **p, int x); //p recebe &L
```

Questão 2

Faça uma função que remove todas as ocorrências (se existir) de um elemento **x** em uma **lista duplamente ligada** (**No *L**).

```
1 void remove_lista_dupla(No **p, int x); //p recebe &L
```

Utilize a seguinte estrutura para implementar listas duplamente ligadas:

```
1 typedef struct no{
2     int valor;
3     struct no *prox, *prev;
4 } No;
```

Questão 3

Um pequeno exército se viu rodeado certa vez por um exército mais forte que ele. A única chance para não serem esmagados seria que alguém fosse buscar reforço montado no único cavalo da tropa.

Para decidir quem seria o sortudo a ir buscar ajuda, decidiu-se colocar todos os soldados em um círculo, sorteando-se então um nome de um soldado e um número M.

A partir do soldado sorteado, o M-ésimo soldado no sentido horário seria retirado da roda tendo que ficar no campo de batalha.

Procedendo-se desta forma, o último soldado que restasse no círculo seria aquele que iria buscar ajuda.

Faça um programa que implementa uma **lista ligada circular** para resolver o problema descrito acima.

Questão 4

Marque como verdadeira (V) ou falsa (F) cada uma das afirmações abaixo. Justifique sua resposta apenas caso a sentença seja falsa.

- (a) () TADs auxiliam na reutilização e manutenção de código.
- (b) () A implementação de um TAD fica na interface (arquivo `.h`).
- (c) () Toda lista encadeada é dinâmica.
- (d) () Listas duplamente encadeadas facilitam o percurso de seus elementos.
- (e) () Um TAD sempre disponibiliza a maneira como as informações foram estruturadas.

Questão 5

O que é um arquivo Makefile?

Questão 6

Um *deque* (*double-ended queue*()) é uma estrutura de dados com as operações: `insere_inicio()`, `insere_fim()`, `remove_inicio()`, `remove_fim()`.

Implemente um TAD para *deque* utilizando listas ligadas.

Questão 7

Faça um TAD que representa um conjunto de inteiros e que suporte as operações mais comuns de conjunto como adição, união, interseção, etc

Questão 8

Faça um TAD que representa uma string. Implemente as operações de comparação, concatenação e cópia.

Questão 9

Faça um TAD que representa um ponto com coordenadas (x, y) . Implemente as operações de criação do ponto, remoção do ponto, e distância euclidiana entre pontos.

Questão 10

Faça um TAD que representa um vetor de números inteiros e que suporte as operações de produto escalar e produto vetorial.