

## 1ª. Lista de Exercícios “Desafio”

**Elaborada por:** Fernando Alva Manchego (estagiário PAE)

1. Sejam  $L$  e  $P$  duas listas ligadas que contêm inteiros, ordenados ascendentemente. A função  $ImprimePosições(L,P)$  imprimirá os elementos em  $L$  que estão nas posições especificadas por  $P$ . Por exemplo, se  $P = 1,3,4,6$ , o primeiro, terceiro, quarto e sexto elemento em  $L$  devem ser impressos. Implemente a função indicada usando só as operações básicas de listas. (Tradução livre de [1]).
2. Dadas duas listas ligadas, combine seus nós para criar uma nova lista cujos nós alternem entre aqueles das duas primeiras listas: <nó 1 da lista 1, nó 1 da lista 2, nó 2 da lista 1, nó 2 da lista 2,...> (Tradução livre de [3]).
3. Uma cadeia só com parênteses () e [] é “correta” se:
  - (a) Se é uma cadeia vazia.
  - (b) Se A e B são corretas, AB é correta,
  - (c) Se A é correta, (A) e [A] são corretas.

Escreva um programa que tome uma sequência de cadeias de caracteres do tipo indicado e verifique se são corretas ou não (Tradução livre de [4]).

4. Implemente o TAD *Conjunto* utilizando listas ligadas. Considere a definição dada em aula.
5. Desejam-se realizar as operações matemáticas básicas (soma, subtração, multiplicação e divisão) sobre números inteiros tão grandes que não podem ser representados por nenhum tipo de dado em C. Assim, a ideia é representar cada número como uma lista ligada simples, onde cada nó contém um dígito. Implemente um TAD para poder manipular este tipo de números e realizar as operações básicas indicadas.
6. Um estacionamento possui só uma linha para um máximo de 10 carros. Só existe uma entrada/saída em um lado da linha. Se um cliente chega para pegar um carro que não é o mais próximo da saída, todos os carros que bloqueiam o seu caminho são movidos, o carro do cliente é retirado e os outros carros são devolvidos na mesma ordem em que estavam originalmente. Escreva um programa que processe um grupo de linhas de entrada. Cada uma delas contém um “C” para chegada e um “S” para saída, e um número identificador do carro. Assume-se que os carros vão chegar e sair na ordem especificada pela entrada. O programa deveria imprimir uma mensagem toda vez que um carro chega ou sai. Quando um carro chega, a mensagem deve especificar se existe vaga no estacionamento. Se não existe vaga, o carro não entra no estacionamento. Quando um carro sai, a mensagem deve incluir o número de carros que foram movidos para que ele possa sair do estacionamento (Tradução livre de [2] com adaptações).

7. Implementar um sistema de inscrição de pessoas para um evento, levando em consideração o seguinte:
- Existem quatro funcionalidades: *cadastro* de pessoas novas, *atualização* dos dados de uma pessoa, *eliminação* de uma pessoa e *listagem* de pessoas cadastradas.
  - Para realizar o *cadastro*, cada pessoa deve fornecer obrigatoriamente os seguintes dados: nome, sobrenome e data de nascimento (dia, mês e ano). Logo, o novo registro deve ser inserido na lista de pessoas cadastradas. Deve-se verificar que não existam registros repetidos.
  - Para realizar a *atualização* de dados, o usuário deve indicar, pelo menos, algum dado fornecido no *cadastro* (nome, sobrenome e/ou data de nascimento). O sistema deve procurar o registro, mostrar o conteúdo e solicitar ao usuário os dados que deseja atualizar. No caso em que o usuário tenha indicado um dado que não permite identificar de maneira única o registro que deseja atualizar (por exemplo, só indica Sobrenome = “Silva” e existem várias pessoas com esse sobrenome, mas com os outros dados diferentes), o sistema deve solicitar mais dados até conseguir identificar a pessoa que deseja atualizar dados.
  - Para realizar a *eliminação*, o processo é similar ao realizado para a *atualização*. Depois de identificar o registro que se deseja eliminar, o sistema deve pedir confirmação ao usuário antes de proceder à eliminação do registro.
  - A funcionalidade de *listagem* deve mostrar na tela a informação de todas as pessoas cadastradas (uma coluna por cada dado) em ordem ascendente por nome. Em caso de empate, deve-se usar o sobrenome para identificar qual registro vá antes, e em caso de um novo empate, utilizar a data de nascimento. Lembre que não existem registros repetidos; portanto, sempre será possível encontrar uma ordem entre os registros usando os três dados.

Utilizar listas ligadas para implementar o sistema. Todas as implementações não devem possuir uma complexidade maior do que  $O(n)$ .

8. O sistema de atendimento numa agência de um banco utiliza duas categorias para determinar na ordem em que os clientes que chegam devem ser atendidos. Toda pessoa que possui uma conta no banco tem a categoria A, caso contrário tem a B. Apresentam-se as seguintes regras para determinar qual cliente deve ser atendido:
- Se existe algum caixa vazio, o cliente (seja A ou B) pode ser atendido imediatamente.
  - Se chegar um cliente da categoria A, ele deve ser atendido na próxima vez que um caixa fique vazio. Se existem outros clientes A esperando, o novo cliente deverá esperar a que eles sejam atendidos primeiro.
  - Um cliente da categoria B só poderá ser atendido se um caixa estiver disponível e se não existe nenhum cliente A esperando.

Implemente um programa para simular esta situação, considerando o seguinte:

- (a) O programa receberá por linha de comando (parâmetros do *main*) o número de clientes A, o número de clientes B e o número de caixas a serem considerados na simulação.
- (b) O programa deverá escolher, aleatoriamente, o cliente que chega em um determinado momento no banco.
- (c) O programa deverá escrever na tela o que está acontecendo na simulação. Exemplo: “chegou cliente A1”, “atendendo cliente B1 no caixa 2”, etc.

### **Referências**

[1] Weiss, M. (1995). Data Structures and Algorithm Analysis. The Benjamin/Cummings Publishing Company.

[2] Langsam, Y.; Augenstein, M.; Tenenbaum, A. (1990). Data Structures Using C and C++. Prentice Hall

[3] [http://www.engr.mun.ca/~theo/Courses/ds/linked\\_lists.html](http://www.engr.mun.ca/~theo/Courses/ds/linked_lists.html)

[4] <http://uva.onlinejudge.org/external/6/673.html>