F7 – Simulação da Prova 1

Atividade avaliativa para cômputo de Frequência
Estruturas de Dados 1 (1001502) - ENPE 4 – 2022

Descrição
- Proponha solução para a prova abaixo. Individual.

O que entregar (em arquivo único – PDF ou DOC, sem compactar)
- Solução para as questões.

Onde entregar
- No ambiente de interação da disciplina no Google Classroom, no link indicado.

Quando entregar
- No prazo indicado no ambiente de interação da disciplina.

Orientações Gerais
- **Tempo para elaboração da prova: 2h**.

Orientações Quanto a Notação, Nomes das Variáveis, e Estruturas
- Use os **mesmos nomes fornecidos no enunciado** (L, F, X, etc.). Utilize variáveis auxiliares temporárias, o tanto quanto for necessário. É só declarar e usar. Mas **não considere a existência de nenhuma outra variável permanente**, além das definidas no enunciado. Não considere prontas para uso nenhuma operação, salvo se explicitamente indicado no enunciado da questão.
- Considere as **estruturas exatamente conforme definido no enunciado**, seja no texto da questão, seja nos diagramas.
- Para o desenvolvimento de algoritmos, use preferencialmente a notação adotada no Livro Texto: **p = NewNode; Deletenode(P), P->Info e P->Next,** sendo P uma variável do tipo **NodePtr** (ponteiro para nó). Quando a estrutura for duplamente encadeada, ao invés de P->Next considere que a notação contenha **P->Dir e P->Esq.** Também é possível implementar em C ou **C++**.

Questão 1 (4 pontos) Considere o Tipo Abstrato de Dado **FILA**, implementado através de uma **lista duplamente encadeada, circular,** segundo os diagramas abaixo. Implemente, **da forma mais apropriada para proporcionar portabilidade e reusabilidade**, as operações Retira, Vazia e Destrói:

- **Boolean Vazia**(variável por referência F do tipo Fila)

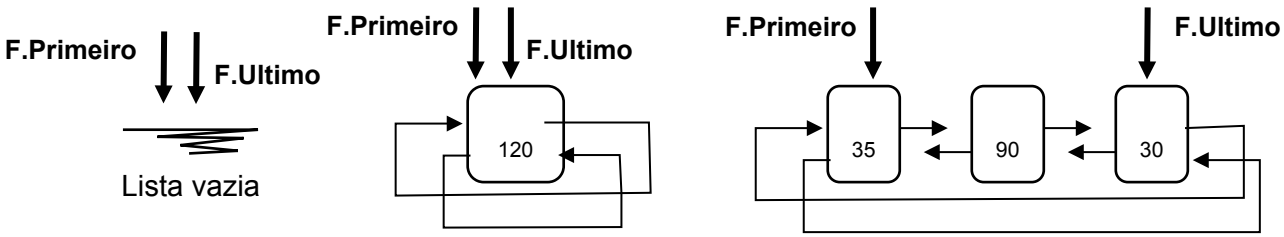
// deve retornar verdadeiro se a fila não tiver nenhum elemento; falso caso contrário.

- **Retira**(variável por referência F do tipo Fila, variável por referência X do tipo Elemento, variável por referência Erro tipo boolean)

// retira 1 elemento da fila F. Erro deve retornar verdadeiro se a fila não tiver nenhum elemento para ser retirado; falso caso contrário.

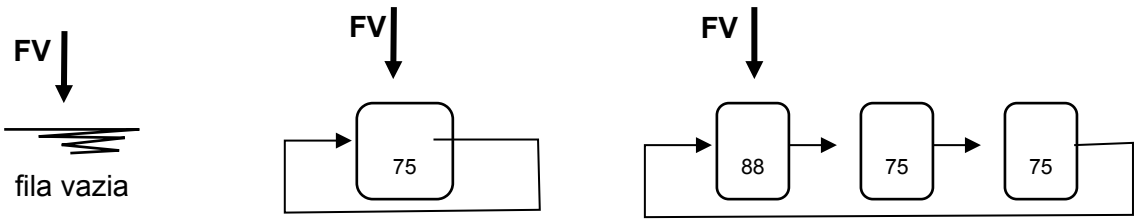
- **Destroi**(variável por referência F do tipo Fila)

// desaloca (remove) todos os elementos da fila.



{o tipo Fila é um registro com 2 campos: F.Primeiro e F.Ultimo, ambos do tipo NodePtr (Ponteiro para nó) }

Questão 2 (4 pontos) Fila de Vacinação (FV). Um município está implementando uma **Fila de Vacinação**. O primeiro critério para entrar na Fila de Vacinação é a **idade**, ou seja: pessoas mais velhas entrarão na fila à frente de pessoas mais jovens que já estiverem na fila. O segundo critério da Fila de Vacinação é a ordem de chegada, ou seja: se já existem pessoas na fila com determinada idade, em anos, novas pessoas com essa mesma idade em anos devem ser inseridas na fila após as pessoas de mesma idade que entraram na fila primeiro. Ou seja, a Fila de Vacinação será uma "Fila de Prioridades", na qual deve ser priorizado o atendimento às pessoas de maior idade em anos. Uma possível solução a essa situação é implementar a Fila de Vacinação através de uma **lista encadeada, circular, ordenada em ordem decrescente pela idade em anos, com elementos repetidos,** conforme os diagramas abaixo.



Nos diagramas, **FV** é um ponteiro para o início da estrutura de dados. A estrutura vazia é composta pelo ponteiro FV apontando para NULL. Implemente a operação:

- **Insere** (variável por referência **FV** do tipo FilaDeVacinacao; variável **Idade** do tipo inteiro);

  /* insere 1 pessoa com a idade fornecida como parâmetro na fila FV. Desconsidere a possibilidade de a estrutura estar cheia. Tipo FilaDeVacinacao = um ponteiro do tipo NodePtr (ponteiro para nó). Deve-se inserir as pessoas mais velhas antes das mais jovens. Pessoas que tenham a mesma idade em anos, devem ser inseridas após as demais que já estão na fila */

Roteiro para a solução:

- Identifique os casos a serem tratados;
- Desenhe cada caso a ser tratado;
- Faça um algoritmo para cada caso a ser tratado, e desenhe passo a passo, até a situação final;
- Faça um algoritmo geral o mais simples possível, abrangendo todos os casos.

Questão 3 (2 pontos) Considere o jogo Spider Shopping: no início o usuário toma conhecimento de uma lista de compras, e deve memorizá-la. Em uma segunda fase, uma série de produtos aparecem em uma vitrine, e o jogador precisa "comprar" (selecionar) somente os itens que constam da lista original de compras. Ao final, são computados os itens comprados corretamente e os itens comprados erroneamente.

Considere prontas para uso as seguintes operações de uma Lista Cadastral:

- Cria(L); // Cria uma Lista Cadastral L, iniciando sua situação como vazia;
- Vazia(L); // Verifica se a Lista Cadastral L está vazia, retornando true para vazia, e false caso contrário;
- Insere (L,X,Ok); // Insere o elemento de valor X na Lista L.O parâmetro Ok deve retornar true se a operação foi bem sucedida; false caso não;
- Retira(L,X,Ok); // Retira da Lista L o elemento de valor X, caso X estiver na Lista. Neste caso, o parâmetro Ok deve retornar true; false caso contrário;
- Boolean EstaNaLista (L, X); // Verifica se o elemento de valor X faz parte da Lista Cadastral L, retornando true caso X estiver na Lista L, e false caso não;
- PegaOPrimeiro(L, X, TemElemento); // X retorna o valor do primeiro elemento da Lista L, se esse primeiro elemento existir. Se não existir esse primeiro elemento (Lista vazia), o parâmetro TemElemento retornará false. Se existir, retornará true;
- PegaOPróximo(L,X,TemElemento); // X retorna o valor do próximo elemento da Lista, em relação à última chamada à uma das operações PegaOPrimeiro ou PegaOPróximo. Se não existir esse próximo elemento (final da Lista), o parâmetro TemElemento retornará o valor Falso. Se existir, retornará true.

Com base em chamadas a estas operações primitivas de uma Lista Cadastral, desenvolva a operação:

Int ItensCompradosErroneamente(ListaCadastral Carrinho de Compras, ListaCadastral ListaDeComprasOriginal);
// Calcula e retorna o número de itens comprados erroneamente, ou seja, itens que constam do carrinho de compras e não constam da lista original.

Considere que o elemento das listas sejam do tipo "Elemento".