Lista 02

Exercícios:

1. **Que é um Tipo Abstrato de Dados (TAD) e qual a característica fundamental na sua utilização?**

é uma especificação de um conjunto de **dados** e operações que podem ser executadas sobre esses no main principal.

1. **Faça a especificação de um sistema de controle de reservas de um clube que aluga quadras poliesportivas usando TAD.**

O cliente irá verificar se possui quadras disponíveis e quantidade de quadras. Quando decidir reservar uma quadra, essa quantidade de disponíveis irá subtrair uma e alocar em quadras reservas.

Só é possível reservar uma quadra por cliente.

1. **O que significa alocação sequencial de memória para um conjunto de elementos?**

Organizar os elementos sequencialmente um após o outro. Determinada como Lista Linear.

1. **O que significa alocação estática de memória para um conjunto de elementos?**

Todo programa ao ser executado aloca um espaço necessário na memória, alguns programas exigem que mais espaços sejam alocados durante o processo do código.

Na alocação estática, o compilador resguarda um espaço na memória fixo, não sendo ampliar essa reserva durante o processo do código.

1. **Qual a diferença entre alocação sequencial e alocação encadeada?**

Alocação sequencial: os elementos processados são alocados na sequência física da memória. E por sabermos qual sua ordem e tamanho, torna-se rápida a procura.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |

Alocação encadeada: casa elementos armazena o endereço do próximo elemento. Para acessar uma informação especifica, devemos percorrer toda a estrutura.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **INFOMAÇÃO** | 3 | 1 | 5 | 4 | 2 |
| **ENDEREÇO** | 4 | 5 | 2 | 3 | 1 |

1. **Quais as vantagens de se utilizar alocação encadeada para um conjunto de elementos? Quais as possíveis desvantagens?**

Vantagens: inserção de novos nós. Pois as informações não precisam ser colocadas sequencialmente na nova lista.

Desvantagens: desperdício de memória e tempo de busca.

1. **Projete uma estrutura do tipo lista sequencial ordenada de 20 espaços para abrigar os inimigos ativos de um jogo de computador em ordem crescente de dificuldade (com relação ao número de pontos de vida). Deve ser possível adicionar ou remover um inimigo da lista à medida que este nasce ou é morto pelo jogador, através das funções INSERIR, REMOVER e LISTAR.**

Código realizado

1. **Desenvolva uma função (com parâmetros) para testar se uma lista F1 tem mais elementos do que uma lista F2 (não se esqueça de mexer nas filas apenas através de seus operadores primitivos.**

Código realizado

1. **Implemente uma lista em um vetor circular, sem armazenar o número total de elementos (sugestão: nunca deixe que o indicador “fim” alcance o indicador “início”, ainda que seja necessário perder uma posição do vetor.**

Código realizado

1. **Projete uma estrutura de lista encadeada para abrigar os IDs dos usuários ativos em uma certa aplicação. A estrutura de um nó da lista é a seguinte: Node( int userId; Node pProx;) Ao final do projeto, seu programa deve rodar as seguintes funções:**

**a) Escreva a função inserir, que cria e insere um nó na primeira posição da lista encadeada. Para simplificar o projeto, não se preocupe em casos de inserção de um ID repetido.**

Código realizado

**b) Escreva a função printLista, que mostra na tela todos os IDs dos usuários ativos num dado momento.**

Código realizado

**c) Escreva a função remover, que exclui um certo ID de usuário da lista e mostra na tela o ID removido.**

Código realizado