

Listas

1. Faça uma função recursiva que retorne o número de nós contidos em uma lista L. Considere que L é simplesmente encadeada, sem nó cabeça e não circular. Os elementos de L não estão necessariamente em ordem.
2. Considere uma lista onde cada nó é composto pelo código, nome e preço de um produto. A lista está ordenada por ordem crescente do código do produto.
 - a. Considerando alocação sequencial estática, faça a declaração desta estrutura e uma função que receba esta lista como parâmetro e que escreva os dados de todos os produtos contidos na lista.
 - b. Considerando uma lista simplesmente encadeada dinâmica, faça a declaração desta estrutura e uma função que calcule e retorne a média de preços.
3. Faça uma função recursiva que escreva se um determinado elemento pertence ou não a uma lista. Considere que L é simplesmente encadeada, sem nó cabeça e não circular.
4. Fazer um programa que crie duas listas simplesmente encadeadas L1 e L2 que guardem números inteiros. Tais listas representarão dois conjuntos distintos. Implementar as operações de interseção, união, diferença e pertinência.
 - Pode ser criado um menu na tela com as quatro operações disponíveis no programa. • As operações devem funcionar da mesma maneira que funcionam quando são considerados conjuntos. Por exemplo, quando for feita a operação de união entre L1 e L2 a lista resultante não poderá conter elementos repetidos.
 - Além das funções de interseção, união, diferença e pertinência podem ser criadas outras funções de acordo com a necessidade.
5. Considere duas listas encadeadas L1 e L2, onde cada nó da lista possui duas partes: dado, que guarda um número inteiro e prox, que guarda o endereço do próximo nó. Para cada item abaixo, faça uma função que receba as duas listas como parâmetros e as concatene de acordo com as características especificadas.

Para concatenar L1 com L2 basta juntar o final de L1 com o começo de L2, formando uma única lista com endereço inicial em L1.

 - a. L1 e L2 são simplesmente encadeadas, não circulares e sem nós cabeça;
 - b. L1 e L2 são simplesmente encadeadas, circulares e sem nós cabeça;
 - c. L1 e L2 são simplesmente encadeadas, circulares e com nós cabeça;
 - d. L1 e L2 são simplesmente encadeadas, não circulares e com nós cabeça;

e. L1 e L2 são duplamente encadeadas, circulares e com nós cabeça;
Pilhas e Filas

1. Faça uma função que receba uma pilha P como parâmetro e que escreva todos os seus elementos. Considere uma pilha de números inteiros. Considere alocação encadeada dinâmica.
2. Faça uma função que receba uma pilha de inteiros P como parâmetro e que passe seus elementos pares para uma pilha chamada PAR, e seus elementos ímpares para uma pilha chamada IMPAR. PAR e IMPAR também devem ser parâmetros da função. Considere alocação encadeada dinâmica.
3. Seja uma fila F de números inteiros com alocação sequencial estática, faça a declaração desta estrutura e uma função que verifique se o elemento do começo da fila é par.
4. Sabendo que um DEQUE é um tipo especial de lista linear onde as inserções e remoções de elementos podem ser realizadas nos dois extremos da estrutura, crie uma função para remoção de elementos. Considere alocação sequencial estática, onde o DEQUE pode guardar no máximo N elementos.
 - Faça o desenho da estrutura do DEQUE D, onde sejam especificadas as variáveis de controle: início, fim e total.
 - A função possuirá como parâmetros: D (o DEQUE) e as suas variáveis de controle (início, fim e total) e POS (que indica a posição de onde o elemento será removido: I - início e F - final).
 - Deve ser usada a idéia de dequees circulares (similar às filas circulares)
5. Faça um programa que simule a ordem de execução de vários processos em uma CPU. • Declare uma fila encadeada que guarde em cada nó o número do processo e seu tempo de execução. O tempo de execução é estimado em unidades inteiras.
 - Preencha a fila com alguns processos.
 - O processo que vai executar é o primeiro da fila.
 - O primeiro processo deve ser retirado da fila e deve ser escrita na tela a mensagem: Executando processo #.
 - Um processo só pode executar por um período máximo de 2 u.t. Caso o seu tempo estimado para execução seja maior do que 2, o processo deve ser inserido ao final da fila novamente.
 - Cada vez que um processo executa o seu tempo deve ser devidamente decrementado.
 - O programa termina a execução quando não existirem mais processos na fila.
 - Crie procedimentos para inserir e remover processos da fila.