

Data de Entrega: 21/11/2018, impresso e em sala de aula. Entregar apenas os códigos das funções.

Considere nesta lista que os seguintes tipos foram definidos:

```
typedef struct { char nome[20];  
                char endereço[40];  
                int id;  
} tipoDados;
```

- 1 – Implemente funções para realizar as operações de criar, inserir e remover elementos em uma pilha implementada com listas encadeadas dinâmicas. Considere que os dados armazenados na pilha são do tipoDados.
- 2 – Implemente funções para realizar as operações de criar, inserir e remover elementos em uma fila implementada com lista encadeada dinâmica. Considere que os dados armazenados na fila são do tipoDados.
- 3 - Implemente funções para realizar as operações de criar, inserir e remover e atualizar ponteiro para atual em uma lista circular implementada com lista encadeada dinâmica. Considere que os dados armazenados na lista circular são do tipoDados. Considere que elementos novos são inseridos depois do atual.
- 4 – Faça uma função que receba como parâmetro um número K, uma lista encadeada circular onde cada nó contenha dados do tipoDados e um ponteiro para o próximo elemento, denominado prox. A função deve retornar o número de elementos da lista com valor de id igual a K. (P 2006)
- 5 – Faça uma função que receba como parâmetro uma pilha vazia e um vetor de inteiros contendo K posições. A função deve inserir na pilha cada valor encontrado no vetor. A ordem de inserção deve ser do primeiro para o último elemento do vetor. Os dados da pilha neste caso são do tipo int. (P 2006)
- 6 – Faça uma função que receba como parâmetros uma lista encadeada com dados do tipoDados e um valor k. A função deve remover o elemento da k-ésima posição da lista. Por exemplo, se k= 3, então a função deve remover o terceiro elemento da lista. Caso não haja k elementos na lista, a função não executa a tarefa.(P 2008)
- 7 – Considere uma lista encadeada dinâmica com dados do tipo inteiro (int). Diz-se que uma lista encadeada é simétrica (conceito inventado para a questão) se cada elemento que ocorre antes da posição do meio (na primeira metade, desconsiderando o elemento do centro se houver número ímpar de elementos), tem um correspondente de valor igual na segunda metade da lista (há um outro com o mesmo valor na segunda metade da lista). Considere que nenhum valor ocorre mais de duas vezes na lista. Como exemplo, as listas <3,6,9, 3,9,6> e <6, 3,9,3,9,6> são simétricas. Já as listas <3,6,9,3,6,5> e <3,6,5,5,2,3> não são simétricas. (P 2008)

8 – Faça uma função para transferir todos os elementos de uma lista encadeada dinâmica (com elementos do tipoDados) para um vetor. Considere que o vetor comporta todos os elementos da lista. É importante que o espaço de memória usado na lista dinâmica seja liberado ao final da função. (P 2008)

9 - Faça uma função para mostrar os elementos de uma fila implementada como lista encadeada. Considere que cada nó da fila é composto de um campo nome, declarado como um vetor de 50 caracteres. A função deve mostrar os elementos na ordem inversa de ocorrência na fila e a fila deve ser implementada utilizando lista encadeada simples, com ponteiros ligando cada nó ao próximo elemento da fila. A fila deve ser percorrida uma única vez na função (dica: recursividade). (P 2006)

10 – Implemente o algoritmo de ordenação QUICKSORT, com chamadas para porções menores que 10 elementos executando o algoritmo de ordenação por inserção. Considere que o vetor a ser ordenado é de elementos do tipoDados, com chave de ordenação sendo o campo nome. Considere ainda a possibilidade de haver elementos repetidos no vetor a ser ordenado.

11 – Implemente o algoritmo de ordenação mergesort para ordenar um vetor de elementos do tipoDados. A ordenação deve ser feita pelo nome.

12 – Implemente o algoritmo de ordenação quicksort para ordenar um vetor de elementos do tipoDados. A ordenação deve ser feita pelo id.

13 – Implemente uma versão modificada do quicksort da questão 2 para ordenação parcial, onde deseja-se obter os K menores elementos de um vetor com N elementos. Considere que será fornecido sempre com valor menor que $K < N$.

Para as questões de 14 a 17, considere os seguintes tipos para a definição de listas encadeadas.

<pre>typedef struct tipoNo { int dado; tipoNo *prox; } tipoNo;</pre>	<pre>typedef struct tipoLista { tipoNo *prim; } tipoLista;</pre>
--	--

14 – Faça uma função que receba como parâmetro uma lista e retorne a soma dos valores presentes na lista encadeada. A função deve retornar zero para listas vazias.

15 – Faça uma função que retorne 1 (verdadeiro) caso uma lista encadeada passada como parâmetro tenha nós com valores estritamente crescentes (elemento da posição $i <$ elemento da posição j , para todo $i < j$). A função deve retornar 0 (falso) em caso contrário. Considere que listas vazias e com 1 elemento são estritamente crescentes.

16- Faça uma função que destrua uma lista encadeada passada como parâmetro, apagando todos os elementos da lista. A função não pode deixar qualquer nó da lista alocado na memória.

17 – Corrija a função de inserção em uma lista encadeada descrita no código abaixo. A função deve inserir um elemento passado como parâmetro em uma lista encadeada também passada como parâmetro. Apresente o código corrigido e explique cada uma das alterações realizadas de maneira a ajudar o programador que cometeu os erros. Você deve realizar o mínimo possível de alterações e apenas as mudanças com devidas explicações serão aceitas. A resposta será considerada completamente incorreta se a função corrigida apresentar qualquer erro.

```
void inserir(tipoLista L, int *d) {  
    tipoNo *aux;  
  
    aux->dado = d;  
    aux->prox = L->prim;  
    L->prim = aux;  
}
```