

**Usando somente as funções do TAD Lista e do TAD Pilha, resolva as questões abaixo:**

1) void invLista(Lista lst)

Inverter uma lista utilizando como apoio uma pilha. A complexidade desta rotina é  $O(N)$ , onde  $N$  é o número de elementos da lista.

2) Pilha copiarPilha( Pilha p )

Fazer uma cópia de uma pilha, usando como apoio uma lista. Também é possível usar como apoio uma pilha. A pilha original deve ser restaurada.

3) void invPilha( Pilha p)

Inverter o conteúdo de uma pilha.

4) int iguaisPilhas( Pilha p1, Pilha p2)

Dizer se duas pilhas são iguais sem destruir seu conteúdo.

5) int avaliarExpressao( Lista lstExp)

Avaliar uma expressão aritmética na notação posfixa.

Nesta notação o operador vem depois dos operandos.

Ex:  $3\ 5\ +\ 10\ *$  tem como resultado 80. Pois,  $3\ +\ 5 = 8$  e  $8\ *\ 10 = 80$

Ver nos slides de Waldemar Celes este exemplo.

Dicas: será preciso usar uma pilha de inteiros, e as funções isdigit e atoi.

A Lista lstExp é uma lista cujos elementos são strings( char \*):

Ao percorrer a lista e ler um elemento desta, tem-se:

- char \* elem;

-elem = infoLista(lstExp);

Para verificar se o elemento lido é um dígito basta fazer:

- if ( isdigit( elem[0] )

- se não for um dígito então assume-se que é um operador.

Para converter uma string(char \*) em inteiro basta fazer:

-atoi(elem)

6) void imprimePilha(Pilha p)

Imprimir o conteúdo de uma pilha, sem alterar seu conteúdo( a pilha deve ser restaurada)

7) void fundoPilha( Pilha p, TipoP elem)

Colocar no fundo da pilha o elemento elem

8) int longPilha( Pilha p)

Computar e retornar o número de elementos da pilha, sem alterar seu conteúdo( a pilha deve ser restaurada)

9) void elimTodosPilha( Pilha p, TipoP elem)

Eliminar da pilha todas as ocorrências do elemento elem.

10) int somaPilha( Pilha p)

Somar todos os elementos da pilha e retorna seu resultado

11) int palindromePilha( Pilha p)

Retornar true(1) se o conteúdo da pilha é um palíndrome e false(0) caso contrário

