

INF 213 - Listas encadeadas

Objetivo: praticar ponteiros, entender melhor conceitos de listas e praticar recursividade.

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR <<--

Arquivos fonte e diagramas utilizados nesta aula:

https://drive.google.com/open?id=1L_9jgaUmzAOT-5GQbWsO3URlfe3qOgUb

Etapa 1

O arquivo MyList.h contém a implementação parcial da lista encadeada vista em sala de aula. Porém, parte do código foi removido. Sua tarefa consiste em completar o código e testar a classe utilizando o programa TestaMyList.cpp .

Obs: implemente a função destroy() utilizando recursividade (dica: você pode criar funções auxiliares (tais funções devem ser públicas ou privadas?))

Utilize um Debugger de memória (<http://valgrind.org/docs/manual/quick-start.html>) para verificar se sua implementação não contém erros de memória.

Etapa 2

Reimplemente a função destroy(), mas desta vez sem utilizar recursividade (não apague a versão anterior da sua função -- apenas a comente).

Etapa 3

Adicione uma função chamada eraseMatchingElements a sua classe (essa função não está disponível nas listas disponibilizadas pela STL). Tal função deverá receber um argumento e remover da lista todos elementos iguais a esse argumento. Ao final deve-se retornar quantos elementos foram removidos (a capacidade da lista não deve ser alterada).

Por exemplo, se o MyList v armazena caracteres e v=[a,b,c,b,w,z], então após a chamada v.eraseMatchingElements('b') o valor de v deverá ser [a,c,w,z] e a chamada da função deverá ter retornado o número 2 (visto que dois caracteres 'b' foram removidos de v).

Etapa 4

Implemente uma função chamada reverse(), que inverte a ordem dos elementos na lista. Para praticar o uso de apontadores, **NÃO** utilize iteradores ou outras funções já providas pela classe MyList.

Dica: use recursividade!

Submissao da aula pratica:

Submeta apenas o arquivo MyList.h contendo a implementacao das suas funcoes.

A solucao deve ser submetida ate as 18 horas da proxima segunda-feira utilizando o sistema submittty (submittty.dpi.ufv.br). Atualmente a submissao so pode ser realizada dentro da rede da UFV.