

sqtpm

[ggm]

[ajuda](#)[voltar](#)**Trabalho:** 57-listas

Data de abertura: 03/05/2025 23:00:00

Data limite para envio: 30/05/2025 18:00:00 ([aberto](#))

Número de arquivos a enviar: entre 1 e 3

Número máximo de envios: 30

Número de envios feitos por ggm: 1

Último envio: 26/05/2025 16:51:40 0%Casos-de-teste abertos: [casos-de-teste.tgz](#)

Recursos: cpu 1 s, memória 32768 kb, stack 8192 kb

Enviar:

Linguagem:

C ▾

Arquivos:

Browse...

No files selected.

Enviar

Operações em listas encadeadas

Escreva um programa para realizar operações em uma lista encadeada de doubles.

Entrada

A entrada é composta por uma sucessão de comandos, um por linha. Neste trabalho deve-se considerar que uma lista encadeada com n nós tem a cabeça na posição 0 e tem o rabo na posição $n - 1$. A posição n refere-se ao fim da lista. Deve-se considerar também que os limites para o tamanho da lista são 0 e 4294967294.

Os possíveis comandos são:

- c

Criar uma lista encadeada vazia. Se o programa já estiver operando uma lista encadeada então ela deve ser liberada da memória e uma nova lista vazia deve ser criada.

- i p x

Inserir x na posição p da lista encadeada. O double x passará a ocupar a posição p na lista encadeada. O double que ocupa a posição p antes da inserção ficará na posição $p + 1$.

Se p for maior ou igual ao tamanho da lista encadeada então a inserção deve ser feita no fim da lista encadeada. Por exemplo, se uma a lista encadeada tiver 5 nós, a inserção de 3.14 na

sqtpm

[ggm]

[ajuda](#)[voltar](#)

posição 9 deverá fazer com que 3.14 seja inserido na posição 5

.

- $r\ p$

Remover um nó da posição p da lista encadeada.

Se não houver um nó naquela posição da lista encadeada, o programa não deverá fazer nada.

- p

Imprimir a lista encadeada em uma única linha. Cada elemento da lista encadeada deverá ser seguido de um espaço. A lista encadeada vazia deverá ser impressa como uma linha vazia.

- $v\ i\ t$

Reverter a ordem dos elementos no trecho $[i..t]$ da lista encadeada. Seja n o tamanho da lista antes da operação. Se $0 \leq i \leq t < n$ então a operação deve reverter a ordem dos elementos nas posições entre i e t inclusive. Caso contrário a operação não deve fazer nada.

- $x\ i\ t\ p$

Mover o trecho $[i..t]$ da lista encadeada para a posição p .

Seja n o tamanho da lista antes da operação. Se $0 \leq i \leq t < n$ então a operação deve mover todos os elementos nas posições entre i e t inclusive de tal forma que o nós movidos fiquem imediatamente antes do nó na posição p . Caso contrário a operação não deve fazer nada.

- t

Terminar. Antes de terminar, o programa deve liberar a memória ocupada pela lista encadeada.

A seqüência de comandos sempre começa com um comando c e sempre termina com um comando t .

Saída

A saída tem as linhas produzidas pelas operações p .

Exemplos

Entrada	Saída
c	0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80
i 0 0.8	0.30 0.40 0.00 0.10 0.20 0.50 0.60 0.70 0.80
i 0 0.7	0.30 0.40 0.00 0.10 0.20 0.50 0.80 0.70 0.60

sqtpm

[ggm]

[ajuda](#)[voltar](#)

i	0	0.6	0.00	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80
i	0	0.5	0.10	0.20	0.30	0.40	0.50	0.60	0.70		
i	0	0.4									
i	0	0.3									
i	0	0.2									
i	0	0.1									
i	0	0.0									
p											
x	0	2	5								
p											
v	6	8									
p											
v	6	8									
x	2	4	0								
p											
r	0										
r	7										
p											
t											

Requisitos adicionais

- Não pode haver qualquer variável global. Uma variável é global se estiver declarada fora de alguma função (variáveis declaradas dentro da main não são globais, são locais à função main).

Sugestões

- Comece fazendo um programa que apenas lê a entrada e imprime. Depois adicione as demais funcionalidades.
- Teste cada funcionalidade separadamente.
- Pode ser interessante criar uma biblioteca para listas com arquivos c e h, de forma similar à organização do trabalho 06.

Sobre organização do código e comentários

- Faça um programa organizado, bem indentado e que seja fácil de ler.
 - Adicione comentários que vão ser úteis para entender o programa se você for relê-lo daqui a alguns anos: comentar cada linha vai ser redundante; documentar blocos de código e a estratégia usada na solução vai ser muito útil.
-