

**sqtpm**

[256352]

[voltar](#)**Trabalho:** 08-lista-simetrica

Linguagens: C

Data de abertura: 2019/10/02 10:00:00

Data limite para envio: 2019/10/09 23:59:59 (encerrado)

**Último envio:** 100% em 2019/10/08 22:07:25

Envios: 3

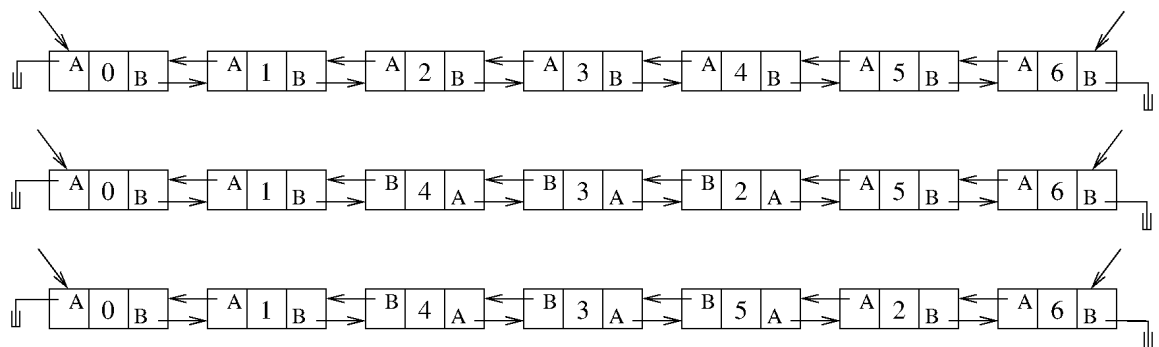
## Lista simétrica

Para reverter um trecho de uma lista encadeada é necessário atualizar todos os apontadores dos nós no trecho, realizando um número de operações proporcional ao número de nós no trecho revertido. Se as operações de reversão forem frequentes, o custo dessas operações se torna alto.

A lista duplamente encadeada também não permite simplesmente reverter um trecho usando um número constante de operações por que, se fizermos isso e tentarmos percorrer a lista, os apontadores *prev* e *next* ficam inconsistentes ao longo da lista. Para resolver o problema temos que reverter de fato o trecho, gastando um número de operações proporcional ao número de nós no trecho revertido.

Nas listas simétricas, que são listas duplamente encadeadas, os apontadores são chamados simplesmente de A e B, sem referência a uma ordem na lista. Para percorrer a lista simétrica é necessário usar informação sobre o nó visitado anteriormente para decidir qual é o próximo nó a ser visitado. Assim, só é possível percorrer a lista a partir da cabeça ou do rabo, não é possível percorrer a partir de um nó qualquer. Uma reversão só precisa testar e ajustar os apontadores que estão entre os limites do intervalo revertido. Nenhum nó interno ao intervalo revertido é atualizado.

Por exemplo considere a lista simétrica abaixo. Na sequência das figuras acontece uma reversão dos nós nas posições entre 2 e 4 e depois uma reversão entre as posições 4 e 5. Observe que apenas os apontadores nos nós dos extremos da parte revertida são atualizados. Observe também que na terceira lista o caminho para percorrer da cabeça para o rabo passa por apontadores BBAAABB.



Com essa restrição é possível reverter um trecho da lista delimitado por dois nós quaisquer fazendo apenas um número constante de operações. Isso é, dados dois nós, o número de operações necessárias para a reversão é constante, já que um número fixo de apontadores é atualizado.

A lista simétrica é uma alternativa eficiente nas aplicações em que a operação de reversão é frequente.

Neste trabalho, implemente uma lista simétrica de double com reversão.

### Entrada

Cada linha da entrada contém um comando como descrito a seguir.

-Inserir:

i posição valor

Inserir um novo nó em uma posição específica. O double inserido passa a ocupar aquela posição na lista, contando da cabeça, que está na posição zero. Se não for possível inserir na posição específica, então a inserção deve ser feita no fim da lista. Essa situação acontece, por exemplo, quando a lista tem 5 elementos e pedimos para

**sqtpm**

[256352]

[voltar](#)

Essa situação acontece, por exemplo, quando a lista tem 5 elementos e pedimos para inserir 3.14 na posição 9. Nesse caso, 3.14 deve ser inserido na posição 5.

- Imprimir:

p

Imprimir os valores na lista a partir da cabeça em uma única linha. Cada elemento da lista deve ser seguido de um espaço. A lista vazia é impressa como uma linha em branco, sem espaços.

- Reverter:

v início término

Reverter o trecho delimitado pelas posições início e término, inclusive.

- Terminar:

t

**Saída**

A saída deve conter as linhas produzidas pelos comandos 'p'.

**Exemplo**

Entrada:

```
i 0 1.2
i 1 2.3
i 2 3.4
i 3 4.5
i 4 5.6
i 5 6.7
i 6 7.8
i 7 8.9
p
v 1 5
p
v 4 7
p
```

Saída:

```
1.2000 2.3000 3.4000 4.5000 5.6000 6.7000 7.8000 8.9000
1.2000 6.7000 5.6000 4.5000 3.4000 2.3000 7.8000 8.9000
1.2000 6.7000 5.6000 4.5000 8.9000 7.8000 2.3000 3.4000
```

---