

**sqtpm**

[thiago]

[voltar](#)**Trabalho:** 03-gerente

Linguagens: C

Arquivos-fonte a enviar: 3.

Data de abertura: 2017/08/21 14:00:00

Data limite para envio: 2017/08/27 18:00:00

Número máximo de envios: 25

Casos-de-teste abertos: [casos-de-teste.tgz](#)

---

## Gerente do Espaço



É bem verdade que a maioria das pessoas não se importa muito com o que ocorre dentro de um computador, desde que ele execute as tarefas que devem ser desempenhadas. Existem, no entanto, alguns poucos nerds que sentem prazer em acompanhar o movimento de bits e bytes dentro da memória do computador.

E para esse público, constituído principalmente de adolescentes, que a multinacional de software ACM (Abstractions of Concrete Machines) deseja desenvolver um sistema que acompanhe e produza um relatório das operações efetuadas em um disco rígido.

### Problema:

Um disco rígido é composto de uma sequência de células atômicas de armazenamento, cada uma de tamanho 1Kb.

Especificamente, a ACM deseja acompanhar três tipos de operações:

- **insere NOME T**  
insere no disco o arquivo NOME, de tamanho T. Você pode supor que um arquivo com esse nome não existe ainda no disco. O tamanho T de um arquivo é dado na forma XKb, XMb, ou XGb, onde X é um inteiro ( $0 < X \leq 1023$ ). NOME é uma cadeia de caracteres com comprimento máximo 10.
- **remove NOME**  
remove o arquivo NOME do disco. Se um arquivo com esse nome

## sqtpm

[thiago]

[voltar](#)

- **otimiza**  
compacta o disco, deslocando os arquivos existentes na direção do início do disco, eliminando espaços livres entre dois arquivos subsequentes, e preservando a ordem em que os arquivos aparecem no disco, de modo a deixar um espaço de memória livre no final do disco.

A capacidade de um disco é sempre um número múltiplo de 8Kb. No início, o disco está vazio, ou seja, contém um bloco livre do tamanho da capacidade do disco. Um arquivo é sempre armazenado em um bloco de células de armazenamento contíguas. O arquivo a ser inserido deve ser sempre colocado no início do menor bloco livre cujo tamanho é maior ou igual ao tamanho do arquivo. Se mais de um bloco livre é igualmente adequado, escolha o mais próximo do começo do disco. Caso não seja possível inserir o arquivo por falta de um bloco livre suficientemente grande, deve-se executar automaticamente o comando **otimiza**. Se após a otimização ainda não for possível inserir o arquivo, uma mensagem de erro deve ser produzida. No caso de todas as operações serem executadas sem erro, seu programa deve produzir uma estimativa aproximada do estado final do disco, conforme descrito abaixo.

Lembre que 1Mb corresponde a 1024Kb, enquanto 1Gb corresponde a 1024Mb.

## Especificação de entrada e saída

### Entrada

A entrada é constituída de vários casos de teste. A primeira linha de um caso de teste contém um único inteiro  $N$  indicando o número de operações no disco ( $0 < N \leq 10000$ ). A segunda linha de um caso de teste contém a descrição do tamanho do disco, composta por um inteiro  $D$  ( $0 < D \leq 1023$ ), seguido de um especificador de unidade; o especificador de unidade é uma cadeia de dois caracteres no formato Kb, Mb ou Gb. Cada uma das  $N$  linhas seguintes contém a descrição de uma operação no disco (insere, remove ou otimiza, conforme descrito acima). O final da entrada é indicado por  $N = 0$ .

### Saída

Para cada caso de teste seu programa deve produzir uma linha na saída. Se todas as operações de inserção forem executadas sem erro, seu programa deve produzir uma linha contendo uma estimativa aproximada do estado do disco, apresentada como se segue. Divida o número de bytes do disco em oito blocos contíguos de mesmo tamanho. Para cada um dos oito

## sqtpm

[thiago]

[voltar](#)

daquele bloco, e apresentar a estimativa do estado final no formato

[C][C][C][C][C][C][C][C]

onde C é ' ', '-' ou '#', dependendo se  $75 < P \leq 100$ ,  $25 < P \leq 75$  ou  $0 \leq P \leq 25$ , respectivamente. Caso um arquivo não possa ser inserido por falta de espaço, seu programa deve produzir uma linha contendo a expressão ERRO: disco cheio; nesse caso, operações subseqüentes do caso de teste devem ser ignoradas.

---

### Exemplo de entrada

```
3
8Kb
insere arq0001 7Kb
insere arq0002 3Mb
remove arq0001
6
8Mb
insere arq0001 4Mb
insere arq0002 1Mb
insere arq0003 512Kb
remove arq0001
remove arq0001
insere arq0001 5Mb
0
```

### Exemplo de saída

```
ERRO: disco cheio
[#][#][#][#][#][#][-][ ]
```

---

### Observações

- O Trabalho deve ser resolvido utilizando listas encadeadas.
  - A definição da lista e as funções que manipulam a lista devem estar em um par de arquivos {c,h} separado do programa principal. Cada submissão deve ter 3 arquivos.
-