

Atividade de Estruturas de Dados II (Listas Encadeadas)

Gerente de 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.

É 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. 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 não existe, não faz nada;
- `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.

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 anteriormente). O final da entrada é indicado por $N = 0$.

A entrada deve ser lida da entrada padrão.

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
```

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 blocos seu programa deve verificar a porcentagem P de *bytes* livres 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 subsequentes do caso de teste devem ser ignoradas.

A saída deve ser escrita na saída padrão.

Exemplo de Saída

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

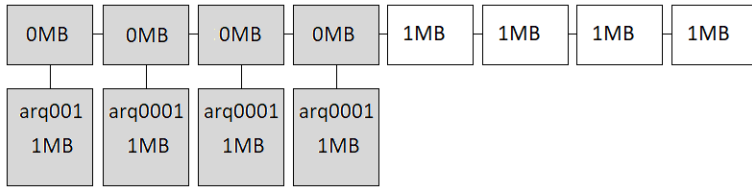
Mais Exemplos de Entrada e Saída:

Entrada	Saída
6	[#][][][#][][][][]
8KB	[#][#][][][][][][]
insere arq1 2KB	[#][][][][][][][]
insere arq2 1KB	[][][][][][][][]
insere arq3 1KB	[][][][][][][][]
remove arq1	ERRO: disco cheio
remove arq2	
insere arq4 1KB	
4	
16MB	
insere a1 2MB	
insere a2 512KB	
insere a3 2MB	
remove a2	
4	
16MB	
insere a1 2MB	
insere a2 513KB	
insere a3 2MB	
remove a2	
24	
16KB	
insere aaaaaaaaaa 1KB	
insere bbbbbbbbbb 1KB	
insere cccccccccc 1KB	
insere dddddddddd 1KB	
insere eeeeeeeeeeee 1KB	
insere ffffffffffff 1KB	
insere gggggggggg 1KB	
insere hhhhhhhhhh 1KB	
insere aaaaaaaaaa1 1KB	
insere bbbbbbbbbb1 1KB	
insere ccccccccc1 1KB	
insere dddddddd1 1KB	
insere eeeeeeeee1 1KB	
insere fffffffff1 1KB	
insere ggggggggg1 1KB	
insere hhhhhhhh1 1KB	
remove aaaaaaaaa1	
remove ccccccccc1	
remove eeeeeeeee1	
remove ggggggggg1	
remove aaaaaaaaaa	
remove ccccccccc	
remove eeeeeeeee	
remove gggggggggg	
3	
8KB	
insere arq0001 7KB	
insere arq0002 3MB	
remove arq0001	
0	

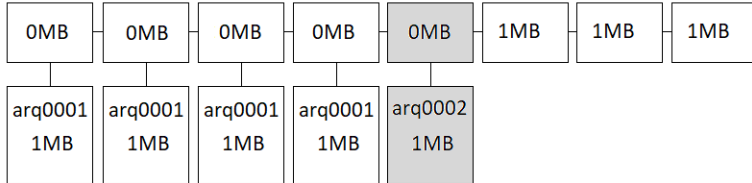
```
6 //número de operações
8MB //cria 8 blocos de 1MB
```



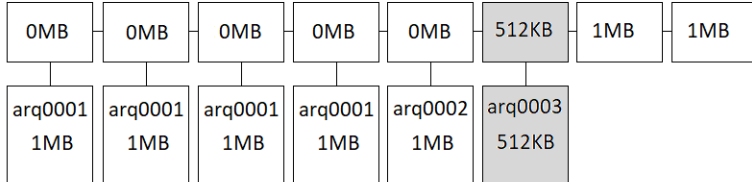
```
insere arq0001 4MB
```



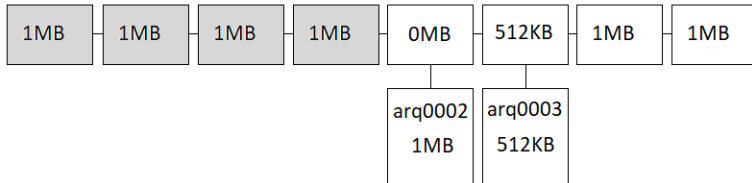
```
insere arq0002 1MB
```



```
insere arq0003 512KB
```

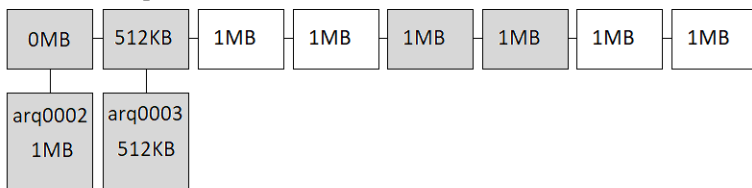


```
remove arq0001
```

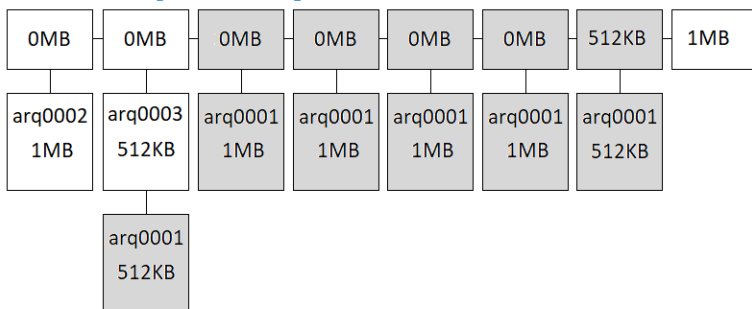


```
remove arq0001 //não faz nada
```

```
insere arq0001 5MB //chama o otimiza
```



```
//insere agora o arq0001 5MB
```



[#][#][#][#][#][#][-][]