



UNIVERSIDADE FEDERAL DO ABC
CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO
Prof. Monael Pinheiro Ribeiro

ESTRUTURAS DE DADOS [Estáticas]

EstruturasDeDados.[c | cpp | java | cs]

Na Ciência da computação, uma estrutura de dados é um modo particular de armazenamento e organização de dados em um computador de modo que possam ser usados eficientemente, facilitando sua busca e modificação.

Diferentes tipos de estrutura de dados são adequadas a diferentes tipos de aplicação e algumas são altamente especializadas, destinando-se a algumas tarefas específicas.

Estruturas de dados e algoritmos são temas fundamentais da ciência da computação, sendo utilizados nas mais diversas áreas do conhecimento e com os mais diferentes propósitos de aplicação. Sabe-se que algoritmos manipulam dados. Quando estes dados estão organizados (dispostos) de forma coerente, caracterizam uma forma, uma estrutura de dados. As estruturas de dados clássicas são: Lista, Fila e Pilha.

Faça um programa que implemente as estruturas de dados estática clássicas, ou seja, Lista Estática Sequencial (LES), Fila Estática, Pilha Estática e Fila Estática Circular.

Entrada

A entrada é composta de vários casos de testes.

Na primeira linha há um inteiro N , $1 \leq N \leq 10^5$, representando a quantidade de estruturas de dados.

Em seguida há N linhas, cada linha conterá as informações sobre cada estrutura de dados, essas informações são apresentadas no seguinte formato:

NOME TAMANHO

Onde: NOME pode ser uma das seguintes palavras: LES, FILA, PILHA ou CIRCULAR. E TAMANHO é um número inteiro, tal que: $1 \leq \text{TAMANHO} \leq 10^5$.

Após as informações sobre as estruturas de dados há uma sequência indefinida de comandos, ou seja, seu término é marcado por final de arquivo (EOF), no seguinte formato:

POS_ED OP CHAVE

Onde:

- **POS_ED** define o índice da estrutura de dados que sofrerá o comando e tem a seguinte restrição: $0 \leq \text{POS_ED} < N$.
- **OP** define a operação que a estrutura de dados sofrerá. **OP** pode ter os seguintes valores: **I** para inclusão de um valor, **M** para exibir os valores da estrutura de dados na tela e **R** para remover uma chave da estrutura de dados.
- **CHAVE** define o valor da chave a ser inserida na estrutura de dados e trata-se de um valor inteiro positivo de 32 bits.

Atenção não haverá a presença de um valor para CHAVE nos seguintes casos:

- Quando $OP = M$;
- Quando $OP = R$ e for uma estrutura de dados do tipo FILA, PILHA ou CIRCULAR.

Vale lembrar também que a estrutura de dados do tipo LES não aceita chaves repetidas.

Além disso, a estrutura de dados do tipo FILA, trata-se de uma fila estática clássica, ou seja, com problema de overflow da fila. Contudo, ao remover o último elemento volte-a para o estado de vazia.

Também ignore qualquer comando de inclusão em uma estrutura de dados cheia, ou uma remoção em uma estrutura de dados vazia. A inclusão de uma chave já existente em uma estrutura de dados do tipo LES também deve ser ignorada.

Saída

A saída é composta de uma quantidade indefinida de linhas, pois dependem dos comandos $OP = M$ e $OP = R$, pois a cada comando de exibir, os valores da estrutura de dados devem ser exibidos, separados por espaço em branco. Atenção após o último elemento e antes do primeiro elemento exibido não há espaço em branco. No caso das estruturas de dados LES, FILA e CIRCULAR os elementos devem ser mostrados do primeiro para o último, no caso da estrutura de dados PILHA devem ser mostrados da base para o topo.

No caso dos comando OP = R, a cada remoção bem sucedida nas estruturas de dados, deve ser impresso na tela o seguinte:

REMOVIDO: CHAVE

Onde, CHAVE assumirá o valor da chave removida da estrutura de dados.

Após cada comando OP = R e OP = M deve-se saltar uma linha, inclusive o último comando.

Exemplos

Entrada	Saída
4	10 50 80
LES 10	50 10 80
FILA 5	50 10 80
PILHA 7	50 10 80
CIRCULAR 8	REMOVIDO: 50
0 I 50	10 80
1 I 50	REMOVIDO: 50
2 I 50	REMOVIDO: 80
3 I 50	REMOVIDO: 50
0 I 10	10 80
1 I 10	50 10
2 I 10	10 80
3 I 10	
0 I 80	
1 I 80	
2 I 80	
3 I 80	
0 M	
1 M	
2 M	
3 M	
0 R 50	
0 M	
1 R	
2 R	
3 R	
3 M	
2 M	
1 M	