1. 1-Inicia função inserir do tipo void, recebendo um ponteiro do tipo fila e uma variável int;

2-Alocação de espaço na memória, o espaço é alocado é de tamanho no; O ponteiro de no chamado novo recebe o endereço do espaço alocado;

3-A estrutura novo recebe v em info. Se usa -> pois novo é um ponteiro;

4-A estrutura novo recebe NULL em prox; Se usa -> pois novo é um ponteiro;

5- Esse teste serve para indicar se a fila está vazia ou não. Verdadeiro indica não vazia;

6-O prox do último no da fila aponta para NULL, ao adicionarmos um novo elemento da fila devemos atualizar o prox, pois ele já não é mais o último, o prox devera agora apontar para esse elemento que acabou de ser inserido;

7-Essa linha atualiza o endereço armazenada em fim, pois um novo elemento foi adicionado;

8- Esse teste serve para indicar se a fila está vazia ou não. Verdadeiro indica que a fila antes estava vazia;

9-Neste caso ele é o primeiro elemento da fila, logo inicio e fim apontam para o mesmo elemento;

10- Fecha a função;

1. 1-Inicia função remover do tipo int, recebendo um ponteiro do tipo fila;

2-Declaração de um ponteiro do tipo no chamado novo\_ini;

3- Declaração de um ponteiro do tipo no chamado removido;

4-Declaração de uma variável to tipo int chamada valor;

5-Removido recebe o ponteiro do primeiro elemento da fila;

6-Novo-ini recebe o endereço ao qual o primeiro elemento aponta, pois esse elemento que ela aponta será o novo inicio;

7-valor recebe o valor do elemento que esta sendo retirado;

8-O endereço de f->ini é atualizado para apontar para o novo primeiro elemento;

9-Retornar verdadeiro indica que não a mais elementos na fila;

10-Se não há inicio não há fim, logo ambos NULL;

11-O espaço do elemento removido se torna disponível;

12-O valor do elemento removido é retornado;

13-Fecha a função;

1. INÍCIO

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 100 |  |  |
| 100 | 90 |  |
| 90 |  |  |
| 90 | 10 |  |
| 10 |  |  |