

Questão 1

Ainda não respondida

Vale 2,50 ponto(s).

Você foi contratado para desenvolver parte do código de um TAD (tipo abstrado de dados) para a disciplina de estrutura de dados. Esse TAD deve fornecer suporte às estruturas de Pilhas e de Filas encadeadas. Desta forma, um programador poderá utilizar o TAD para escrever programas que necessitem de rotinas de Pilha e Fila (tal como no exemplo **principal.c**). Ocorre que você já desenvolveu à algumas semanas um TAD de Pilha (**pilhaE.h** e **pilhaE.c**) que contém todas as estruturas de dados e cujo arquivo **.h** aparece abaixo. Assim, você teve a ideia de, ao invés de criar um TAD de Fila desde a estaca zero, utilizar as funções do TAD de Pilha para **emular** as funções do TAD de Fila e criar um módulo **filasE.c** e **filasE.h** (que internamente utilizam chamadas às rotinas do TAD de Pilha). Um fragmento de código aparece abaixo, complete o arquivo **filasE.c** com o código necessário para implementar rotinas de enfileirar, desenfileirar e imprimir os elementos da fila. Note que você precisa apenas finalizar o arquivo **filasE.c**.

```
1 // pilhaE.h
2 typedef struct Cell Cell;
3 typedef struct PilhaE PilhaE;
4
5 Cell* criar_celula(int key);
6 PilhaE* criar_pilhaE();
7 int pilhaE_vazia(PilhaE *p);
8 void empilhar(int key, PilhaE *p);
9 int desempilhar(PilhaE *p);
10 void imprimir_pilha(PilhaE *p);
11 int liberar_pilha(PilhaE *p);

1 // principal.c
2 #include <stdio.h>
3 #include "filaE.h"
4 #include "filaE.c"
5
6 int main(){
7     int x;
8     FilaE* f;
9     PilhaE* p;
10
11     f = criar_filaE();
12     p = criar_pilhaE();
13
14     enfileirar(7,f);
15     enfileirar(-13,f);
16     enfileirar(18,f);
17     enfileirar(-2,f);
18
19     imprimir_fila(f);
20
21     while(!filaE_vazia(f)){
22         x = desenfileirar(f);
23         printf("x = %d\n",x);
24         empilhar(x,p);
25     }
26
27     imprimir_pilha(p);
28
29     liberar_filaE(f);
30     liberar_pilha(p);
31
32     return 0;
33 }
```

```

1 //filaE.h
2 #include "pilhaE.h"
3
4 #define FilaE PilhaE
5
6 //Protótipo de funções
7 FilaE* criar_filaE();
8 int filaE_vazia(FilaE* f);
9 void enfileirar(int key, FilaE* f);
10 int desenfileirar(FilaE* f);
11 void imprimir_fila(FilaE* f);
12 int liberar_filaE(FilaE* f);

```

```

1 //filaE.c
2 #include "pilhaE.h"
3 #include "pilhaE.c"
4 #include "filaE.h"
5
6 FilaE* criar_filaE(){
7     return criar_pilhaE();
8 }
9
10 int filaE_vazia(FilaE* f){
11     return (pilhaE_vazia(f));
12 }
13 void enfileirar(int key, FilaE* f){
14
15     /* ??? */
16 }
17 int desenfileirar(FilaE* f){
18
19     /* ??? */
20 }
21
22 void imprimir_fila(FilaE* f){
23
24     /* ??? */
25 }
26
27 int liberar_filaE(FilaE* f){
28     return liberar_pilha(f);
29 }

```

◀  Primeira Prova

Seguir para...