

Lista de Exercícios de Estruturas de Dados e Algoritmos

Considere a existência de um tipo abstrato pilha de números inteiros, cuja a interface é definida no arquivo pilha.h da seguinte forma:

```
typedef struct pilha TPilha;  
TPilha * inicializa (void);  
void push (TPilha *p, int elem);  
int pop (TPilha *p);  
void libera (TPilha *p);  
int vazia (TPilha *p);
```

Leve em consideração, também, a existência de um tipo abstrato fila de números inteiros, cuja a interface é definida no arquivo fila.h da seguinte forma:

```
typedef struct fila TFila;  
TFila* inicializa (void);  
TFila* insere (TFila *f, int elem);  
int retira (TFila *f);  
void libera (TFila *f);  
int vazia (TFila *f);
```

Usando somente estas operações, escreva:

(a) uma função que, dada uma fila f, retorne uma pilha, contendo todos os elementos de f, e obedecendo a ordem de entrada dos inteiros na fila f, isto é, o primeiro inteiro que sair da fila f deve ser o primeiro inteiro a sair da pilha. Não é possível alterar a ordem dos elementos da fila de entrada. A função deve obedecer o seguinte protótipo: **TPilha* f2p (TFila *f);**

(b) uma função que, dada uma fila f, retorne uma nova fila, com todos os elementos de f, porém invertida em relação a fila f. Não é possível alterar a ordem dos elementos da fila de entrada. A função deve obedecer o seguinte protótipo: **TFila* inv_fila (TFila *f);**

(c) uma função que, dada uma fila f, retorne uma nova fila contendo os elementos pares de f, e modifique a fila original (f) para que ela possua somente os elementos ímpares, respeitando a ordem de entrada em f. A função deve obedecer o seguinte protótipo: **TFila* sep_fila (TFila *f);**

(d) escreva uma função que, dadas duas filas f1 e f2, retorne uma fila, contendo todos os elementos de f1 seguidos pelos elementos de f2, e obedecendo a ordem de entrada dos inteiros nas filas de entrada. Além disso, as filas de entrada devem permanecer inalteradas. A função deve obedecer ao seguinte protótipo: **TFila* Junta_Filas (TFila *f1, TFila *f2);**

(e) escreva uma função que receba uma pilha e retire todos os elementos pares desta pilha. A função deve obedecer o seguinte protótipo: **void retira_pares (TPilha *p);**

(f) escreva uma função que, dada uma pilha p, retorne uma fila, contendo todos os elementos de p, e obedecendo a ordem de entrada dos inteiros em p, isto é, o primeiro inteiro que sair da pilha p deve ser o primeiro inteiro a sair da fila. Não é possível alterar a ordem dos elementos da pilha de entrada. A função deve obedecer o seguinte protótipo: **TFila* p2f (TPilha *p);**

(g) implemente uma função que receba três filas, f, impares e pares, e separe todos os valores guardados em f de tal forma que os valores pares são colocados na fila pares e os valores ímpares na impares. Ao final da execução desta função, f permanecerá com todos os seus elementos. O protótipo desta função é o seguinte: **void separa_filas(TFila * f, TFila *par, TFila * impar).**