

### Lista de Exercícios de Estruturas de Dados I

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);
```

Q1) Usando somente estas operações, escreva 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\* Fila2Pilha (TFila \*f).**

Q2) Usando somente estas operações, 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 na pilha 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\* Pilha2Fila (TPilha \*p).**

Q3) Implemente uma função que crie uma cópia de uma fila, passada como parâmetro de entrada. A função não deve alterar a fila de entrada e deve ter o seguinte protótipo: **TFila\* Copia (TFila \*f).**

Q4) Implemente uma função que receba três filas, f, impares e pares (todas já inicializadas), 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).**

Q5) Implemente uma função que, dada uma pilha de entrada p, retorne uma pilha com todos os elementos de p ordenados de maneira crescente. Ao final da execução desta função, p permanecerá com todos os seus elementos. O protótipo desta função é o seguinte: **TPilha\* Ordena (TPilha \*p).**