

EXERCÍCIOS - FILAS E PILHAS

1) Implementar a Fila dinâmica:

```
# define MaxFila 8 /* a fila pode guardar ate 8 elementos */
typedef struct{
    int inicio;          /* inicio indica o local do primeiro da fila */
    int fim;             /* fim indica o local do ultimo da fila */
    int item [MaxFila]; /* os elementos são guardados a partir do zero */
} t_Fila;
typedef t_Fila* Fila;

// INTERFACE
Fila criarFilaVazia( );
int acessarInicio(Fila);
bool verificarFilaVazia(Fila);
bool verificarFilaCheia(Fila);
void pushFila(Fila, int);
void popFila(Fila);
void esvaziarFila(Fila);
```

2) Acrescentar ao pacote de operações do TAD Fila a função

`void furaFila(Fila, int)`

que coloca um elemento no início da fila.

2) Implementar um deque (double ended queue) - uma fila com as operações de inserção e remoção permitidas em qualquer uma das duas extremidades.

3) Para resolver as questões seguintes somente é permitido utilizar as interfaces do TAD Pilha e TAD Fila. Implementar a função e fazer o teste no programa principal.

- Implementar uma função para inverter a ordem dos elementos de uma fila.
- Implementar uma função para inverter a ordem dos elementos de uma pilha.
- Resolver o problema de Josephus utilizando uma fila.

INTERFACE - Pilha

```
void criarPilhaVazia(Pilha *);
int acessarTopo(Pilha *);
bool verificarPilhaVazia(Pilha *);
bool verificarPilhaCheia(Pilha *);
void pushPilha(Pilha *, int);
void popPilha(Pilha *);
void esvaziarPilha(Pilha *);
```

<http://www.cut-the-knot.org/recurrence/flavius.shtml>

http://webpace.ship.edu/deensley/DiscreteMath/flash/ch1/sec1_1/josephus.html