

Universidade Tecnológica Federal do Paraná (UTFPR)  
Departamento Acadêmico de Informática (DAINF)  
**Estrutura de dados I**  
Professor: Rodrigo Minetto  
([rminetto@dainf.ct.utfpr.edu.br](mailto:rminetto@dainf.ct.utfpr.edu.br))

## Lista de exercícios - TAD Fila

1) Codifique uma fila para armazenar inteiros em linguagem C. Modifique o programa **exercicio1.c** para adicionar a funcionalidade que insere e remove elementos em uma fila. Esse programa conforme definido na função main deve imprimir os números 1 e 2 para estar correto. Envie pelo moodle quando tiver terminado.

2) Vimos que para implementar um fila é necessário ter um vetor, um início, um fim e um tamanho. As funções para enfileirar e desenfileirar elementos dessa forma devem receber esses elementos. Ou seja,

```
void enqueue (int vetor[], int elemento, int *fim, int tamanho);  
  
int dequeue (int vetor[], int *ini, int tamanho);
```

Um jeito mais elegante de desenvolver um TAD Fila é reunir todos esses elementos em uma estrutura. Ou seja, primeiramente é necessário definir um struct, e dessa forma, do mesmo jeito que é possível declarar um inteiro através de

```
int variavel;
```

podemos com o uso do código abaixo

```
typedef struct _fila {  
    int ini;  
    int fim;  
    int tam;  
    int *vetor;  
} Fila;
```

também declarar uma variável do tipo Fila, por exemplo

```
Fila *fila;
```

Graficamente, a estrutura acima cria uma variável complexa que engloba várias variáveis simples. Ou seja

```

      -----
*fila = |  ini          |
        |  fim          |
        |  tam          |
        |  vetor = [ ... ] |
      -----
```

Para acessar os elementos em \*fila é necessário agora fazer

```
fila->ini = valor;
```

```
fila->vetor[...] = valor;
```

A vantagem deste tipo de estrutura é notável. Veja como são simplificadas as funções enqueue e dequeue mostradas anteriormente:

```
void enqueue (Fila *fila, int elemento);
```

```
int dequeue (Fila *fila);
```

Note o porque de usarmos **Fila \*fila** e não **Fila fila** (sem o asterisco). Você lembra do problema de passagem por parâmetros por valor e por referência?

Com base nas considerações acima, modifique o programa **exercicio2.c** de tal forma a adicionar a funcionalidade desejada para implementar uma estrutura de dados tipo fila com o uso de uma struct. Esse exercício tem a mesma funcionalidade do exercício 1 e deve imprimir os números 1 e 2 para estar correto.