

Disciplina - Estruturas de Dados Lineares
Lista 2 de Exercícios de Implementação do TAD – Stack e Queue

1ª Questão: Em sala de aula, implementamos o código do enqueue (enfileirar) do TAD Queue (fila) de inteiros para uma fila definida conforme código abaixo. Esta fila utiliza, em sua definição, lista encadeada.

```
typedef struct NoQueue {  
    int info;  
    struct NoQueue* prox;  
}TNoQueue;  
  
typedef TNoQueue* Queue;
```

Termine a implementação do código que iniciamos em sala (código anexo a esta tarefa). Implemente os métodos que faltam.

2ª Questão: Implemente um programa para converter um número, fornecido pelo usuário na notação decimal (base 10), para um número expresso no sistema de numeração binário (base 2). A conversão é realizada pela divisão repetida, pela base, à qual o número está sendo convertido, e então se tomando os restos da divisão na ordem inversa. Por exemplo, a conversão para binário do número 6 exige três de tais divisões: $6/2 = 3$ com resto **0**, $3/2 = 1$, com resto **1** e, finalmente, $1/2 = 0$ com resto **1**. Os restos 0, 1 e 1 são colocados na ordem inversa, de modo que o binário equivalente a 6 é igual a 110.

OBS.: A solução do problema proposto é bem simples se utilizarmos, em nossa lógica, o TAD Stack (pilha). Implemente uma solução para o problema proposto utilizando uma **pilha ENCADEADA** de inteiros para armazenar os restos das divisões.

3ª Questão: Implemente um programa para solicitar ao usuário que digite uma palavra (String) e determinar se a palavra digitada é um palíndromo ou não, ou seja, se a palavra é soletrada identicamente do início para o fim e do fim para o início. Exemplos de palíndromos: “asa”, “ovo”, “radar”, “reviver”.

OBS.: A solução do problema proposto é bem simples se utilizarmos, em nossa lógica, o TAD Stack (pilha). Implemente uma solução para o problema proposto utilizando uma **pilha ENCADEADA de caracteres** para armazenar os caracteres que compõem a palavra sendo analisada.

4ª Questão: O banco SóDinheiro está querendo implantar um sistema de controle de senhas automático. A ideia é a seguinte: quando o cliente chegar ao banco, ele se dirige a um terminal de computador e solicita uma senha de atendimento. De posse de sua senha, que é um número que corresponde a sua ordem de chegada, o cliente pode se sentar e esperar por sua vez de ser atendido. Os funcionários do banco atendem os clientes de acordo com a ordem das senhas emitidas. Toda vez que um funcionário chama uma senha para atendimento, a senha é retirada da fila dos clientes esperando atendimento. Implemente um programa para fazer o controle das senhas de atendimento utilizando uma fila de inteiros para armazenar as senhas. O programa (aplicação) deverá ter as seguintes funcionalidades:

- (a) Um **procedimento** para fornecer uma senha de atendimento a um cliente. Este procedimento utiliza o método enqueue para inserir a senha fornecida ao cliente na fila de espera.
- (b) Um **procedimento** para chamar uma senha para atendimento. Este procedimento utiliza o método dequeue para remover da fila de espera a senha do cliente que será o próximo a ser atendido. Pode acontecer que, em um determinado momento, não haja clientes esperando atendimento. Caso isso ocorra, o procedimento deve emitir uma mensagem informando que não há clientes na fila de espera.

A aplicação deverá utilizar uma **Queue (Fila) ENCADEADA** para armazenar as senhas dos clientes a espera de atendimento.

5ª Questão: Pela lei do consumidor, os clientes devem esperar por atendimento sentados. Como, no banco SóDinheiro, na área dos caixas, só possui espaço para 20 cadeiras, foi estabelecido que pode haver, no máximo, 20 (vinte) clientes dentro do banco esperando pelo atendimento dos caixas. Implemente uma nova versão do programa da questão 4 para satisfazer a esta restrição. Neste caso, se um cliente solicitar uma senha de atendimento e já tiver 20 (vinte) clientes na fila de espera, a senha não será emitida, o

cliente deverá ser informando que não existem senhas disponíveis e que ele deverá retornar em um outro momento.

Aqui também, a aplicação deverá utilizar uma **Queue** (Fila) para armazenar as senhas dos clientes a espera de atendimento. A fila, neste caso, continuará sendo implementada utilizando **lista encadeada**. Porém, neste caso, a fila estará cheia quando tiver 20 clientes esperando por atendimento. **Modifique sua implementação de fila para atender a esta restrição.**