**Exercícios – Filas**

1. Implemente uma rotina que retorne o maior, o menor e a média aritmética dos elementos de uma Fila.
2. Implemente um algoritmo que leia um número indeterminado de valores inteiros. O valor 0 (zero) finaliza a entrada de dados. Para cada valor lido, determinar se ele é um número par ou ímpar. Se o número for par, então incluí-lo na FILA PAR; caso contrário, incluí-lo na FILA ÍMPAR. Após o término da entrada de dados, retirar um elemento de cada fila alternadamente (iniciando-se pela FILA ÍMPAR) até que ambas as filas estejam vazias.
3. Crie uma função que insere elementos ordenados na fila.
4. Implemente um procedimento que reposiciona os elementos na fila de forma que o início se torne fim e vice-versa.
5. Crie um procedimento cuja finalidade seja a criação de uma cópia da fila.
6. Considerando uma fila encadeada (usando ponteiros) faça uma função que a percorra e apresente o maior valor da fila.
7. Implemente para uma Fila Dinâmica Simplesmente Encadeada uma operação cujo objetivo seja a concatenação dessa fila com um outra recebida também, por parâmetro.
8. Mostre a situação da fila F, inicialmente vazia, após a execução de cada uma das operações a seguir:

enqueue(f, a);

enqueue(f, b);

enqueue(f, c);

enqueue(f, f.dequeue(f));

enqueue(f, d)

dequeue(f);

enqueue(f, e);

dequeue(f);

1. Manoel percebeu que seu estacionamento com uma entrada era um fiasco. Ele resolveu vender o seu terreno e comprar um novo que possui uma entrada e uma saída no fundo do terreno. Quando chega um novo carro, este é estacionado no terreno de Manoel, um atrás do outro. Quando um carro precisa sair, os carros do terreno são retirados pela saída, dão uma volta na quadra e são colocados no final da fila pela entrada do estacionamento. Faça um sistema que inclua carros no estacionamento informando o número da placa e retire carros usando o identificador (placa).

Cada vez que se informa uma nova placa, deve ser mostrado o estado do estacionamento (imprimir a ordem em que os carros estão enfileirados). Crie, como exemplo, uma seqüência de entradas e saídas de carros do estacionamento do Manoel.

*Observações importantes:*

* Sua fila deve guardar elementos do tipo String (placa do carro);
* Quando retirar um carro do estacionamento e não for o carro que precisa sair, não se esqueça de incluí-lo novamente no estacionamento.

1. Um hospital de cardiologia precisa de um sistema para efetuar o cadastro de pacientes que necessitam de doação de coração. Para cada paciente que é incluído no sistema deve ser informado o nome, telefone e o grau de urgência para transplante. O grau de urgência é definido na seguinte escala: (3) Urgente; (2) Média urgência; (1) Sem urgência. Sempre que o hospital recebe um novo coração o sistema é consultado para obter o próximo paciente que deverá ser operado. O sistema informa o nome e o telefone do paciente beneficiado pelo transplante. É possível visualizar o tamanho da fila de espera.

Observações:

* Os indivíduos com maior urgência devem ser atendidos primeiro. Para isso, deve-se utilizar três filas diferentes, uma para cada prioridade. Quando chega um coração, verificar se existem pacientes esperando na fila de urgência, se não houver, verifique na fila de média urgência, e assim sucessivamente.
* Para resolver o exercício, crie três rotinas:

1. recebe\_coracao() - retira as pessoas da fila de espera;
2. recebe\_Pacientes() - insere pessoas na fila de espera; e
3. vizualizar\_fila() - verifica quantas pessoas estão esperando por corações.
4. Uma central de atendimento a clientes tem vários atendentes, mas um número muito maior de linhas telefônicas recebendo chamadas. As chamadas são colocadas em uma fila de espera segundo a ordem de chegada (e atendidas quando possível). Ocorre que algumas destas chamadas vêm de longe, e neste caso, se elas ficam esperando na linha, elas ficam causando uma despesa muito maior do que as chamadas que vem de perto. Uma solução alternativa seria colocar as chamadas em fila segundo a prioridade definida primeiramente pelo custo (as mais caras devem esperar menos) e secundariamente por ordem de chamada. Projete uma estrutura para modelar essa situação alternativa.