1 - O algoritmo bubble-sort é considerado um dos algoritmos de ordenação mais simples na computação. Seu pseudo-código é:

```
BUBBLESORT (V[], n)
1     houveTroca <- verdade // uma variável de controle
2     enquanto houveTroca for verdade faça:
3         houveTroca <- falso
4         para i de 1 até n-1 faça:
5             se V[i] vem depois de V[i + 1]
6                 então troque V[i] e V[i + 1] de lugar e
7                 houveTroca <- verdade
```

Usando o conceito de listas apresentado em sala de aula, implemente duas rotinas que: a) ordene um lista usando contiguidade física (vetores) e b) ordene listas definidas por contiguidade lógica (lista encadeada). As rotinas devem receber como parâmetro apenas a lista a ser ordenada e retornar um ponteiro para o primeiro elemento da lista reorganizada em ordem crescente.

2 - Dada duas listas encadeadas ordenadas, implemente uma função que intercale estas em uma única lista ordenada.

3 - Desenvolva uma código que verifique se duas listas são iguais, ou seja, têm mesmo número de elementos e mesmo conteúdo.

4 - Implemente uma função que inverta uma lista encadeada usando apenas uma passagem pela própria lista.

5 - Adicione ao tipo abstrato de dados Lista um campo inteiro que marque a posição do nó na lista encadeada. Desenvolva uma função que encontre de forma automática o elemento localizado no meio da lista. Retorne o ponteiro para este elemento.

6 - 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).

7 - Considere a implementação de filas usando uma lista circular. Escreva uma função FuraFila(TipoFila* pFila, TipoItem x) que insere um item na primeira posição da fila. O detalhe é que seu procedimento deve ser O(1), ou seja, não pode movimentar os outros itens da fila. (observe que este neste caso, estaremos desrepeitando o conceito de FILA – primeiro a entrar é o primeiro a sair).

8 - 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: (5) Muito urgente; (4) Urgente; (3) Médio; (2) Pouco urgente; (1) Sem urgência. Sempre que é 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.

9- Escreva uma função que, dada uma string contendo uma expressão matemática, retorna se ela tem parêntesis agrupados de forma correta, ou seja, se o número de parêntesis à esquerda é igual ao número de parêntesis à direita e se todo parêntese

aberto tem um correspondente à direita, nessa ordem. A expressão ((a+b) viola a primeira condição, mas)a+b(viola apenas a segunda condição. Utilize um pilha para solucionar o problema.

10- Faça uma função que inverta a ordem dos elementos de um pilha.

11- A notação matemática de operações com que estamos habituados é a notação infixa, onde (5 + 4) / 3 significa a divisão da soma dos valores 5 e 4 por 3. Outra notação, chamada pós-fixada, apresenta os valores antes de mostrar a operação a ser realizada entre os mesmos. Assim, a expressão anterior em notação pós-fixada fica 5 4 + 3 /.

Um algoritmo escrito em alto nível para tratar expressões pós-fixadas seria:

enquanto não chegou o final da string
    leia um caracter
    se for um número
        coloque-o na pilha
    senão //é um operador
        desempilhe 2 números e faça a operação determinada pelo caracter que acabou de ser lido
        empilhe o resultado
    fim-se
fim-enquanto
desempilhe o último valor //este é o resultado da expressão

Com uma representação esquemática da pilha para cada passo de iteração, mostre o resultado obtido da expressão:
3 5 2 - * 2 7 2 - + + 4 /

Apresente, ainda, a expressão em notação infixa.

12 -