DISCIPLINA DE ESTRUTURA DE DADOS
Exercícios Revisão – Avaliação A1

1 - LISTA COM CONTIGUIDADE FÍSICA

Considerando as seguintes definições:

```
typedef struct {
    int num;
    char nome[10];
} Contato;
```

```
typedef struct {
    Contato v[MAX_NODOS];
    int n;
} ListaCF;
```

```
#define MAX_NODOS 5
#define SUCESSO 0
#define LISTA_VAZIA 1
#define LISTA_CHEIA 2
#define CONTATO_INEXISTENTE 3
```

Implemente as seguintes operações sobre uma LISTA COM CONTIGUIDADE FÍSICA:

exibeLista
Entrada: uma lista.
Retorno: nenhum.
Descrição: exibe todos os nodos da lista.

consultaContato
Entrada: uma lista e um número de telefone.
Saída: um contato do tipo Contato.
Retorno: SUCESSO, CONTATO_INEXISTENTE ou LISTA_VAZIA.
Descrição: obtém os dados do contato que possui o número de telefone informado.

incluiAntes
E/S: uma lista.
Entrada: um número de telefone e um contato do tipo Contato**.**
Retorno: SUCESSO, CONTATO_INEXISTENTE ou LISTA_CHEIA.
Descrição: inclui um contato antes do contato que possui o número de telefone informado.

excluiContato
E/S: uma lista.
Entrada: um número de telefone**.**
Saída: o contato excluído.
Retorno: SUCESSO, CONTATO_INEXISTENTE ou LISTA_VAZIA.
Descrição: exclui o contato que possui o número de telefone informado.

PILHA COM CONTIGUIDADE FÍSICA

Considerando as seguintes definições:

```
typedef struct {
    int cod;
    float salario;
} Funcionario;
```

```
typedef struct {
    Funcionario v[MAX_NODOS];
    int topo;
} PilhaSE;
```

```
#define MAX_NODOS 5
#define SUCESSO 0
#define PILHA_VAZIA 1
#define PILHA_CHEIA 2
#define CODIGO_INEXISTENTE 3
```

Implemente as seguintes operações utilizando uma PILHA COM CONTIGUIDADE FÍSICA:

empilha
E/S: uma pilha.
Entrada: um funcionário do tipo Funcionario**.**
Retorno: SUCESSO ou PILHA_CHEIA.
Descrição: inclui os dados de um funcionário no topo da pilha.

desempilha
E/S: uma pilha.
Saída: os dados do funcionário armazenados no topo da pilha.
Retorno: SUCESSO ou PILHA_VAZIA.
Descrição: exclui os dados de um funcionário do topo da pilha.

exibePilha
Entrada: uma pilha.
Retorno: nenhum.
Descrição: exibe todos os nodos da pilha.

consultaExistencia
Entrada: uma pilha e um código.
Retorno: SUCESSO, PILHA_VAZIA ou CODIGO_INEXISTENTE.
Descrição: verifica a existência de um funcionário que possui o código passado como argumento.

FILA SIMPLES COM CONTIGUIDADE FÍSICA

Considere as seguintes definições:

```
typedef struct {
    int num;
    char cia[10];
} Voo;
```

```
typedef struct {
    Voo v[MAX_NODOS];
    int frente, re;
} FilaSE;
```

```
#define MAX_NODOS 5
#define SUCESSO 0
#define FILA_VAZIA 1
#define FILA_CHEIA 2
#define NUMERO_INEXISTENTE 3
```

Implemente as seguintes operações utilizando uma FILA COM CONTIGÜIDADE FÍSICA:

insere **E/S:** uma fila. **Entrada:** um vôo do tipo **Voo.** **Retorno:** SUCESSO ou FILA_CHEIA. Descrição: inclui os dados de um voo no topo da pilha.	**retira** **E/S**: uma fila. **Saída**: os dados do voo que está na frente da fila. **Retorno**: SUCESSO ou FILA_VAZIA. Descrição: Retira um vôo da fila.
exibeFila **Entrada**: uma fila. **Retorno**: nenhum. Descrição: exibe todos os nodos da fila.	**consultaExistencia** **Entrada:** uma fila e um número. **Retorno:** SUCESSO, FILA_VAZIA ou NUMERO_INEXISTENTE. Descrição: verifica a existência de um voo que possui o número passado como argumento.

BOM TRABALHO!