



Lista de Exercícios 1

Filas

1. Escreva uma função, em C, para inserir elementos em duas filas (*fila1 e fila2*) implementadas em um mesmo vetor, e uma outra função para retirar elementos desta estrutura, de acordo com os itens abaixo:
 - a) a função para inserir um elemento deverá receber os seguintes parâmetros: o elemento (inteiro) a ser inserido e a indicação da fila (fila1 ou fila2) para inserir tal elemento. Além disso, a função deverá retornar uma mensagem informando se tal elemento foi inserido, ou não;
 - b) a função para retirar um elemento deverá receber como parâmetro a indicação da fila (fila1 ou fila2) para retirar o elemento (inteiro) e, deverá retornar uma mensagem informando se tal elemento foi retirado, ou não.
2. Considere duas filas circulares estáticas armazenadas em um mesmo vetor de tamanho MAX, definido abaixo. Para as filas foram definidas seis variáveis: inicio1, fim1, inicio2, fim2, tam1, tam2, representando os inícios, os fins e tamanhos dessas filas. As filas crescem em sentidos opostos, sendo a primeira fila começando da posição 0 do vetor e podendo ir até a posição 49, e a segunda começando na posição 99 e podendo ir até a posição 50.

```
#define MAX 100;  
int inicio1, inicio2, fim1, fim2, tam1, tam2;  
int vetor_fila[MAX];
```

Escreva uma função, em C, para intercalar essas duas filas, gerando uma nova fila armazenada em uma lista duplamente encadeada dinâmica. Defina a estrutura e as variáveis necessárias para essa nova fila.

3. Considere N filas estáticas, cada uma de tamanho TAM, armazenadas em um vetor de *Filas*. Para tal, foram utilizadas estruturas definidas da seguinte forma:

```
#define N 20  
#define TAM 50  
struct descritor {  
    int inicio;  
    int fim;  
    int tam_fila;  
} DESC[N];  
int FILAS[N*TAM];
```



Portanto, *DESC[0].inicio*, *DESC[0].fim*, *DESC[0].tam_fila* indicam o início, o fim e o tamanho da primeira fila do vetor, respectivamente.

Escreva funções, em C, para realizarem as seguintes operações:

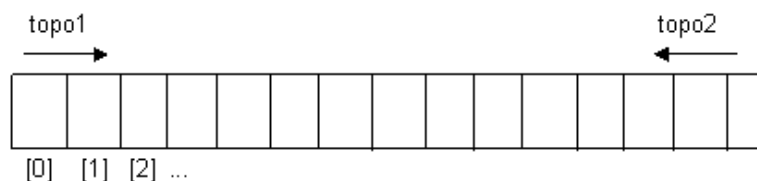
- (i) inicializar todas as filas do vetor;
- (ii) inserir um elemento (*int*) em uma determinada fila;
- (iii) retirar um elemento de uma determinada fila.

As funções para inserir e retirar elementos na fila devem receber como parâmetro um inteiro para indicar a fila na qual será realizada a operação.

Pilhas

- 4. Use as funções *Empilha*, *Desempilha*, *PilhaVazia* para construir outras funções que façam as seguintes operações em uma pilha:
 - a. dado um inteiro *n*, retorne o *n*-ésimo elemento a partir do topo da pilha, deixando a pilha sem seus *n* elementos superiores;
 - b. dado um inteiro *n*, retorne o *n*-ésimo elemento a partir do topo da pilha, deixando a pilha inalterada.
- 5. Escreva funções, em C, para inserir elementos em duas pilhas implementadas em um mesmo vetor. A função deverá receber como parâmetros pelo menos as seguintes informações: a pilha e o elemento (inteiro) a ser inserido. Ao término, ela deverá retornar uma mensagem indicando se tal elemento foi inserido, ou não.
- 6. Considere duas pilhas estáticas armazenadas em um mesmo vetor de tamanho MAX. Para cada pilha foram definidas duas variáveis: *topo1* e *topo2*, as quais representam os topos de cada pilha, conforme a estrutura e a figura abaixo.

```
#define MAX 100;  
typedef struct no {  
    int topo1, topo2;  
    int vetor_pilha[MAX];  
} T_no;
```



vetor_pilha

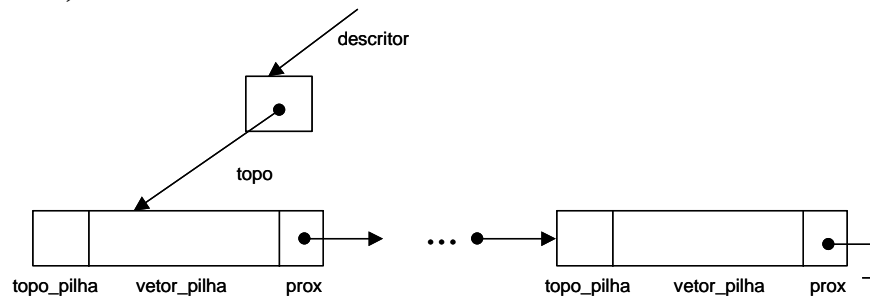
Escreva uma função, em C, para criar uma lista simplesmente encadeada (dinâmica) que contenha elementos intercalados das duas pilhas.

- 7. Considere uma pilha implementada em uma lista simplesmente encadeada (dinâmica). Em cada nó da lista existe uma outra pilha, implementada em um vetor, contendo elementos inteiros. Esta última pilha deve ser implementada em um vetor de tamanho TAM, conforme a definição abaixo:



```
#define MAX 100;
typedef struct no {
    int topo_pilha;
    int vetor_pilha[MAX];
    struct no *prox;
} T_no;
```

```
struct desc {
    T_no *topo_pilha;
} *descricao;
```



Escreva três funções em C para realizar as seguintes operações: (i) inicializar a pilha (dinâmica); (ii) em-pilhar um inteiro com a menor quantidade de elementos; e (iii) desempilhar um inteiro com a maior quantidade de elementos.

OBS.: Não existem pilhas com a mesma quantidade de elementos.

Listas simplesmente encadeadas (dinâmicas)

8. Escreva funções, em C, que recebam uma lista simplesmente encadeada ordenada, e realizem as seguintes operações:

- (i) retirar todos os elementos ímpares que são primos, e, gerar uma outra lista contendo apenas os elementos retirados da lista inicial, em ordem decrescente;
- (ii) retirar todos os elementos pares da lista inicial.

OBS.: Considere que em cada nó são armazenados apenas um elemento (inteiro) e um ponteiro para o próximo nó da lista.

9. Funcionários de uma empresa são representados através da seguinte estrutura:

```
struct func {
    int matricula;
    char nome[20];
    struct data {
        int dia, mes, ano;
        float salario;
    };
};
```



UNIVERSIDADE DO ESTADO DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA
SISTEMAS DE INFORMAÇÃO
ESTRUTURA DE DADOS I

Em cada registro, a estrutura *data* deve conter a data de admissão do funcionário na empresa.

Registros são armazenados utilizando-se uma lista simplesmente encadeada. Escreva funções, em C, para realizar as seguintes operações:

- (a) eliminar todos os funcionários com salário maior do que R\$1.000,00;
- (b) aumentar, em 5%, o salário de cada funcionário com mais de cinco anos na empresa.

10. Considere que os dados dos fornecedores de duas empresas estão organizados segundo uma lista simplesmente encadeada ordenada pelo nome dos fornecedores. Num determinado instante as duas empresas resolveram construir um único cadastro de fornecedores mantendo a mesma estrutura (lista simplesmente encadeada ordenada). Escreva uma função em C que receba os dois cadastros e gere o novo cadastro sem registros duplicados. Para cada fornecedor, considere apenas: o código, o nome e o endereço.

Sugestão: Use a função **int strcmp (const char *s1, const char *s2)** para comparar a string **s1** com a string **s2**. A função retorna 0, menor do que 0 ou maior do que 0 se **s1** for igual a, menor do que ou maior do que **s2**, respectivamente.

11. Considere dois polinômios representados por meio de duas listas simplesmente encadeadas, cujos nós contêm 3 campos: coeficiente, expoente e referência ao seguinte. Usando esta representação, escreva uma função, em C, para somar dois polinômios.

12. Escreva uma função, em C, que receba uma lista simplesmente encadeada cujos nós contêm elementos inteiros em ordem crescente, e retire todos aqueles que são primos. A função deve gerar uma outra lista contendo apenas os elementos retirados da lista inicial, em ordem decrescente.

OBS.: A função não deverá alocar nenhum nó, e todas as funções utilizadas na resolução deverão ser implementadas.

13. Escreva uma função, em C, que receba uma lista simplesmente encadeada representando os alunos de uma determinada escola, e imprima os alunos com a maior e a menor média. Esta lista contém em cada nó os seguintes elementos: (i) o código de matrícula dos alunos (inteiro); (ii) um apontador para uma segunda lista, e; (iii) um apontador para o próximo nó da primeira lista.

A segunda lista contém em cada nó os seguintes elementos: (i) o código da disciplina (inteiro); (ii) um apontador para uma terceira lista contendo as três notas obtidas na disciplina, e; (iii) um apontador para o próximo nó da lista.

A terceira lista contém, em cada nó: o valor de cada nota (real) e um apontador para o próximo nó da lista.



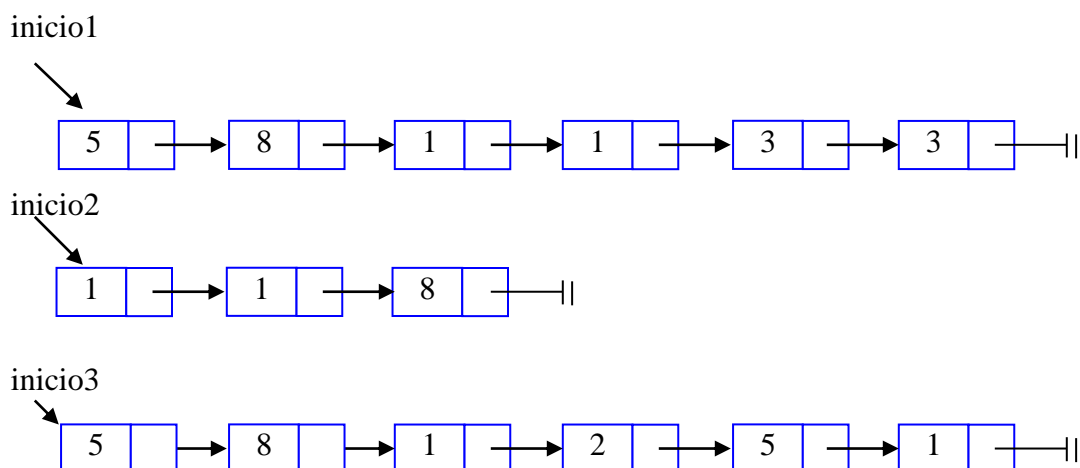
UNIVERSIDADE DO ESTADO DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA
SISTEMAS DE INFORMAÇÃO
ESTRUTURA DE DADOS I

14. Escreva uma função, em C, que receba uma lista simplesmente encadeada, e retorne uma lista ordenada (crescente) sem alocar espaços para novos elementos da lista.
15. Escreva uma função, em C, que receba duas listas simplesmente encadeadas A e B, contendo informações (Código e Nome) dos pacientes do médico X e do médico Y (respectivamente), de uma determinada clínica. A clínica deseja possuir uma lista única com as informações dos pacientes desses dois médicos a partir da junção dessas duas listas. Considere que as listas A e B estão ordenadas por código de pacientes e que não existem dois (ou mais) pacientes com o mesmo código. A lista resultante deve conter todos os pacientes dos médicos X e Y ordenados por código e sem elementos repetidos.

OBS.: A função não deverá alocar nenhum nó, e todas as funções utilizadas na resolução deverão ser implementadas.
16. Escreva uma função, em C, para eliminar todos os elementos inteiros repetidos em uma lista simplesmente encadeada. Considere que os elementos nesta estrutura não estão ordenados.
17. Considere uma lista simplesmente encadeada que armazene números em ordem crescente. Escreva uma função, em C, que receba esta lista, e remova o nó antecessor e o sucessor a um determinado nó contendo um elemento X (inteiro).
18. Considere uma lista simplesmente encadeada que armazene números em ordem crescente. Escreva uma função em C que receba esta lista, e remova o nó antecessor cujo conteúdo é um número par mais próximo a um determinado nó contendo um elemento X (inteiro).
19. Considere uma lista simplesmente encadeada na qual cada nó contém as seguintes informações: matrícula (tipo inteiro) de um professor, ponteiro para o próximo nó dessa lista e um ponteiro para o primeiro nó de uma lista que contém os códigos (tipo inteiro) das disciplinas lecionadas por este professor. Implemente uma função em C para imprimir a matrícula do professor que leciona mais disciplinas.
20. Escreva uma função, em C, que receba uma lista simplesmente encadeada contendo em cada nó um inteiro, e retorne uma outra lista contendo nas primeiras posições os elementos pares em ordem decrescente, seguidos dos elementos ímpares em ordem crescente.
OBS.: A função não deverá alocar nenhum nó, e todas as funções utilizadas na resolução deverão ser implementadas.
21. Escreva uma função, em C, para somar dois números inteiros. Cada número está armazenado em uma lista simplesmente encadeada dinâmica, e cada nó da lista contém



um dígito de um número e um apontador para o próximo nó. O primeiro nó de cada lista contém o dígito mais significativo (dígito mais à esquerda do número) e o último nó de cada lista contém o dígito menos significativo (dígito mais à direita do número). A função deve receber como parâmetros o endereço do início da lista que contém o primeiro número e o endereço do início da lista que contém o segundo (as duas listas já existem). A função deve retornar o endereço do início da lista que conterá o número correspondente à soma dos dois números. Veja o exemplo abaixo que soma os números 581133 e 110:



22. Uma maneira de representar um conjunto é pela lista de seus elementos. Supondo esta representação, escreva uma função, em C, que receba duas listas simplesmente encadeadas e implemente a operação de interseção. O resultado final deve ser apresentado através de uma outra lista simplesmente encadeada (estática).

OBS.: A solução deverá considerar os dois tipos de implementações de listas: estática e dinâmica.

Listas duplamente encadeadas (dinâmicas)

23. Escreva um programa, em C, para inserir, retirar e consultar uma lista duplamente encadeada dinâmica. Após cada operação, o programa deve informar a quantidade de nós, da direita para a esquerda e da esquerda para a direita. (implemente um menu de opções).



UNIVERSIDADE DO ESTADO DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA
SISTEMAS DE INFORMAÇÃO
ESTRUTURA DE DADOS I

24. Considerando uma lista duplamente encadeada (dinâmica) contendo números inteiros, escreva uma função, em C, para eliminar todos os elementos múltiplos de três e retornar uma fila contendo estes elementos (ordenados).
25. Escreva uma função, em C, para ordenar uma lista duplamente encadeada (dinâmica) contendo elementos inteiros. Neste processo de transformação nenhum novo nó da lista resultante deve ser alocado.

Utilize as seguintes estruturas para a solução:

```
struct no {  
    int info;  
    struct no *suc, *ant;}  
  
struct descriptor {  
    struct no *inicio, *fim;} *Lista_inicial, *Lista_final;
```

Lista_inicial é uma variável que aponta para o nó descritor da lista inicial. Tal variável deve ser recebida como parâmetro pela função. **Lista_final** é uma variável que aponta para a lista ordenada. Esta variável deve ser retornada pela função.

26. Escreva uma função, em C, para transformar uma lista duplamente encadeada (dinâmica) em uma outra lista duplamente encadeada, da seguinte forma:
- (i) o primeiro nó deve apontar para o último;
 - (ii) o último nó deve apontar para o segundo;
 - (iii) o segundo deve apontar para o penúltimo nó, e assim por diante.
- Neste processo de transformação nenhum novo nó da lista final deve ser alocado.

Considere também que cada nó da lista e o nó descritor utilizam as seguintes estruturas:

```
struct funcionario {  
    int codigo;  
    char nome[30];  
    struct funcionario *suc, *ant;}  
  
struct descriptor {  
    struct funcionario *inicio, *fim;} *lista_inicial, *lista_final;
```

27. Escreva funções, em C, para implementar a chegada e a saída de 50 documentos em um departamento. Funcionários do departamento são representados por uma lista duplamente encadeada (dinâmica) e o conjunto de documentos para serem avaliados pelo funcionário é representado por uma fila (estática). Documentos que chegam no departamento são inseridos na fila associada ao primeiro funcionário (início da lista). Ao sair da mesa do funcionário, o documento segue para a fila do próximo funcionário até que o mesmo seja avaliado por todos os funcionários do departamento. Em seguida, o documento é retirado do departamento.



UNIVERSIDADE DO ESTADO DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA
SISTEMAS DE INFORMAÇÃO
ESTRUTURA DE DADOS I

As seguintes funções devem ser implementadas:

- (i) inserir documentos na estrutura (documentos que chegam no departamento);
- (ii) retirar documentos do departamento;
- (iii) retirar um documento da mesa de um funcionário.

Para cada funcionário utilize, pelo menos, as seguintes informações: código (**int**) e nome (**char**). Documentos devem ser representados pelos seus códigos (**int**).

28. Escreva uma função, em C, para transformar uma lista duplamente encadeada (dinâmica), contendo uma quantidade ímpar de elementos, em uma outra lista simplesmente encadeada (dinâmica). Tal lista deve conter apenas os elementos das posições ímpares da lista inicial recebida como parâmetro pela função, da seguinte forma:

- (i) o primeiro nó deve apontar para o último;
- (ii) o último nó deve apontar para o terceiro;
- (iii) o terceiro deve apontar para o antepenúltimo nó, e assim por diante.

Neste processo de transformação, os elementos das posições pares devem ser desalocados e, nenhum novo nó da lista resultante deve ser alocado.

Considere também que cada nó da lista e o nó descritor utilizam as seguintes estruturas:

```
struct funcionario {  
    int codigo;  
    char nome[30];  
    struct funcionario *suc, *ant;}  
  
struct descritor {  
    struct funcionario *inicio, *fim;} *Lista_inicial, *Lista_pos_impar;
```

OBS.: **Lista_inicial** é uma variável que aponta para o nó descritor da lista inicial recebida como parâmetro pela função e, **Lista_pos_impar** é uma variável que indica a lista retornada pela função.