

Tipos e Estruturas de Dados
Profa. Graça Nunes
Lista de Exercícios
(Listas Encadeadas, Duplamente Encadeadas e Circulares)

1. Refaça todos os itens do exercício 1 da Lista de Exercícios anterior (Listas Lineares Seqüenciais) considerando que L_1 seja uma **lista ordenada encadeada**.
2. Considere as seguintes declarações:

```
type  rec = record of
        info : integer;
        lig  : indice
    end;
var   A : array[1..n] of rec;
      Dispo, L : indice;
```

Sejam: L, a lista ordenada armazenada da forma **encadeada** em A (L aponta o primeiro registro da lista) e Dispo, a lista encadeada em a que abriga os registros disponíveis. Pede-se os seguintes subprogramas:

- (a) função *numero* (L) – retorna o número de elementos da lista L.
- (b) procedimento *troca* (k, L, v) – modifica o valor do k-ésimo elemento da lista para v.
- (c) procedimento *insere_esq* (k, L, v) – insere o valor v à esquerda do k-ésimo elemento da lista L.
- (d) procedimento *insere_dir* (k, L, v) – insere o valor v à direita do k-ésimo elemento da lista L.
- (e) procedimento *elimina* (k, L) – elimina o k-ésimo elemento da lista L.
- (f) função *busca* (v, L) – retorna “true” se v é um elemento de L; “false”, caso contrário. Faça uma função usando sentinela e outra sem sentinela.
- (g) procedimento *insere* (v, L) – insere (se já não estiver lá) v como elemento de L.
- (h) procedimento *elimina* (v, L) – elimina (se estiver lá) o elemento com valor v da lista L.
- (i) função *banir* (L, v) – elimina todas as ocorrências do elemento v da lista L.

(Obs.: utilize procedimentos para a manipulação da lista Dispo)

3. Considere a implementação estática em Pascal de duas listas encadeadas num mesmo array.

```
type  indice = 0 .. n
      nó = record
        info : integer;
        lig  : indice
    end;
```

```

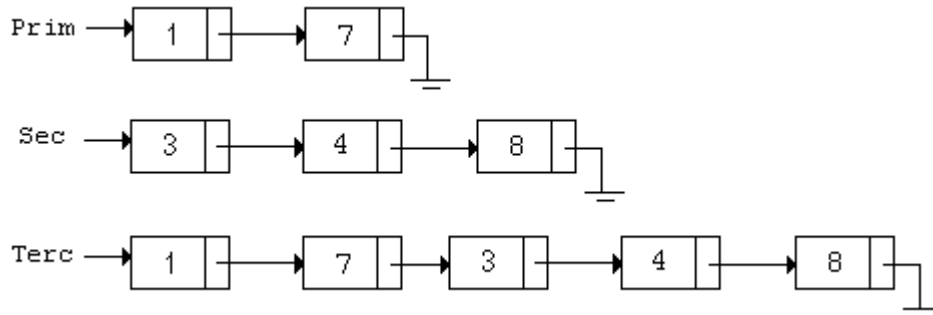
lista = record
    L : array[1..n] of nó;
    Prim, Sec, Dispo : índice
end;

var A : lista;

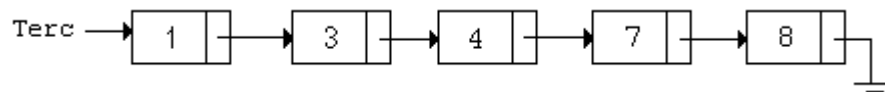
```

Sejam *Prim* e *Sec* “ponteiros” para o primeiro elemento de cada uma delas.

(a) Faça um programa Pascal que concatene as duas listas, dando origem a uma única lista com *Terc* apontando seu primeiro elemento. Exemplo:

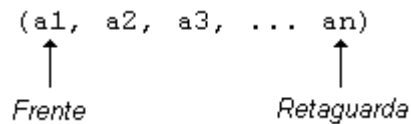


(b) Faça um programa Pascal que intercale as duas listas, dando origem a uma única lista ordenada, com *Terc* apontando seu primeiro elemento. Exemplo:



Obs.: Considere as 2 listas ordenadas crescentemente.

4. Uma “fila de duas pontas” (deque) é uma lista mais geral, onde inserções e eliminações são feitas em ambas as extremidades:



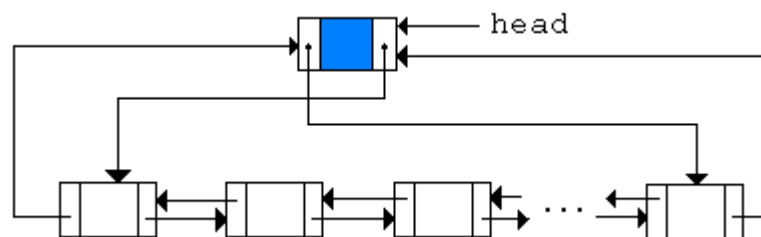
Há duas variações de filas de duas pontas: a fila de duas pontas com “restrição de entrada” – onde inserções são feitas apenas na retaguarda; e a fila de duas pontas com “restrição de saída” – onde eliminações só são permitidas na frente da fila. Faça cada um dos itens a seguir usando:

- (i) representação encadeada estática (array)
- (ii) representação encadeada dinâmica (ponteiros).

(a) construa um algoritmo para inserir um elemento numa fila de duas pontas. Note que um parâmetro deve especificar em que ponta da fila a inserção será feita.

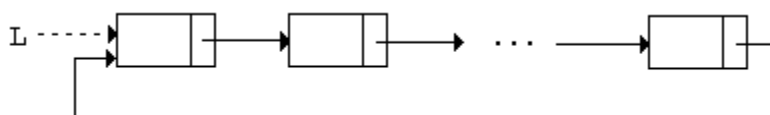
(b) Idem (a) para eliminação.

- (c) Repita (a) e (b) para fila de duas pontas com restrição de entrada.
- (d) Repita (a) e (b) para fila de duas pontas com restrição de saída.
5. Faça um algoritmo que inverta uma lista encadeada, isto é, o último elemento passa a ser o primeiro, o penúltimo passa a ser o segundo, e assim por diante, e o primeiro passa a ser o último. Faça a inversão através da inversão dos campos de ligação, e NÃO dos campos de informação.
 6. Faça um procedimento recursivo para imprimir uma lista encadeada.
 7. Seja uma lista encadeada cujos registros possuam informação do tipo integer. Escreva um programa que ordena a lista em ordem crescente em relação à informação.
 8. Faça um procedimento que remove todas as ocorrências do elemento x de uma lista encadeada.
 9. Dada uma lista encadeada, faça um procedimento que conte o número de elementos desta lista.
 10. Uma **lista duplamente encadeada** possui registros que tem ligações com o sucessor e o predecessor na lista. Ainda, é usual ter um “nó head”, que é um registro auxiliar que aponta para o “primeiro” e o “último” registro da lista e é apontado por eles.

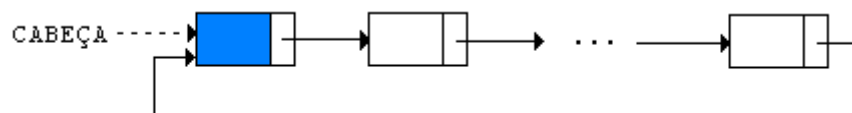


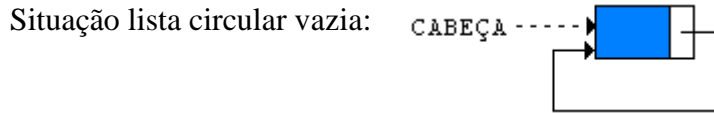
Construa um conjunto de procedimentos para busca, inserção e eliminação de elementos.

11. Escreva um procedimento para concatenar duas listas duplamente encadeadas.
12. Uma **lista encadeada circular** é uma lista encadeada cujo último elemento aponta para o primeiro:



Vantagem: cada elemento é acessível a partir de qualquer outro. Numa lista circular, não faz mais sentido se falar em primeiro ou último elemento. Porém, devemos saber, durante um percurso na lista, se já demos uma volta completa, para evitarmos *loops* infinitos. Para isso, assumimos a existência de um registro especial, chamado **Cabeça de Lista**, cujo campo de informação não pertence ao conjunto de elementos da lista (poderia até servir de sentinela numa busca):





Construa algoritmos para:

- (a) contar o número de elementos numa lista circular;
 - (b) inserir um elemento à esquerda da cabeça da lista;
 - (c) eliminar o elemento de valor x ;
 - (d) concatenar duas listas circulares;
 - (e) intercalar duas listas ordenadas;
 - (f) fazer uma cópia da lista;
13. Generalize a lista circular do exercício anterior para **Lista Circular Duplamente Encadeada**, e repita o itens (a) até (f).
14. Desejamos manipular polinômios do tipo $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$. Tais polinômios podem ser representados por listas simplesmente encadeadas onde cada nó da lista possui três campos: um para o coeficiente que é um número real, um para o expoente que é um número inteiro e um campo que armazena um ponteiro para o próximo nó. Escreva programas para: ler um polinômio e armazená-lo na lista, somar dois polinômios, multiplicar dois polinômios e derivar um polinômio.
15. Suponha que uma lista seja declarada da seguinte forma:

```

type nó = record
    bit : 0..1;
    lig : ^ nó
end;
lista = ^ nó;

```

Um número binário $b_1b_2\dots b_n$, onde cada b_i é 0 ou 1, tem o valor numérico $\sum_{i=1}^n b_i \cdot 2^{n-i}$. Esse número binário pode ser representado pela lista b_1, b_2, \dots, b_n . Essa lista, por sua vez, pode ser representada como uma lista encadeada de nós do tipo nó. Escreva um procedimento **incremento(num_bin)** que adiciona um ao número binário apontado por num_bin. Dica: faça incremento recursivo.

16. Explique o que acontece nas atribuições abaixo (dica: use desenhos):

- (a) $p^{\wedge}.lig := q;$ (b) $p^{\wedge}.lig := q^{\wedge}.lig;$ (c) $p^{\wedge}.info := q^{\wedge}.info;$
- (d) $p := q;$ (e) $p^{\wedge}.lig := nil;$ (f) $p^{\wedge} := q^{\wedge};$
- (g) $p := p^{\wedge}.lig;$ (h) $p := (p^{\wedge}.lig)^{\wedge}.lig;$

17. Simule a execução do algoritmo de Ordenação Topológica visto em sala de aula, para construir a estrutura de dados e fazer uma ordenação topológica para a seguinte ordem parcial:

$$4 < 3 \quad 2 < 5 \quad 7 < 8 \quad 5 < 7 \quad 3 < 8 \quad 4 < 2$$

18. Escreva um operador `AchaKEsimo(int K)` para uma lista encadeada, que recebe uma lista encadeada e devolve uma cópia do k -ésimo maior elemento da lista.

19. Descreva uma implementação de fila dupla utilizando listas encadeadas. Em particular, escreva o operador `RemoveDireita()`.

20. Considere um conjunto implementado através de listas encadeadas. Escreva os seguintes operadores:

- Conjunto União (Conjunto B)
- Conjunto Intersecção (Conjunto B)
- Conjunto Pertence ()
- Conjunto Diferença (Conjunto B)

21. Considere uma lista encadeada em que cada elemento armazena um caractere. Escreva um método `TrocaPalavra(ListaEnc P, ListaEnc Q)` que troca todas as ocorrências da palavra P por Q na lista.

22. Considere a implementação de polinômios utilizando listas encadeadas circulares com cabeça de lista. Escreva um método `Raiz(x):Boolean` que verifica se x é uma raiz do polinômio.

23. Considere o seguinte método para a criptografar mensagens secretas:

- 1ª. etapa: inverter as seqüências de não vogais, incluindo espaços e pontuação.
- 2ª. etapa: inverter a mensagem resultante.

Exemplo:

Dada a mensagem:

ESTRUTURAS DE DADOS É MUITO LEGAL.

Após a primeira etapa, teremos:

ERTSUTURAD SED ADO SÉM UITOL EGA.L

E depois da segunda etapa teremos:

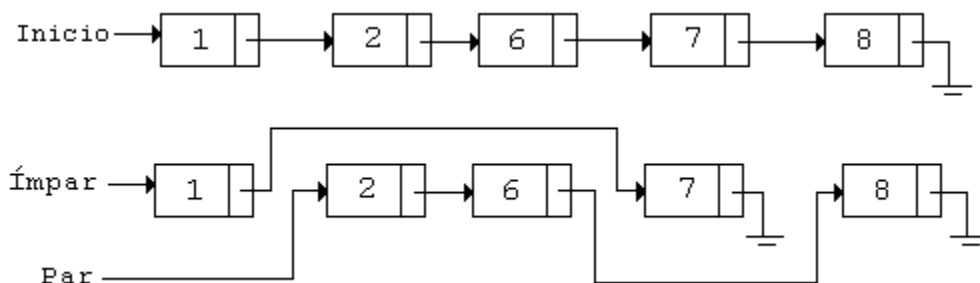
L.AGE LOTIU MÉS ODA DES DARUTUSTRE

Dada uma lista linear encadeada, na qual cada posição contém um caractere da mensagem codificada, faça um programa que obtenha a mensagem original na mesma lista.

24. Escreva funções para (a) remover e (b) inserir um elemento num vetor ordenado. Compare sua rapidez e eficiência com as funções correspondentes para listas encadeadas.

25. Escreva funções para as seguintes operações:

- (a) acrescentar um elemento ao começo de uma lista encadeada;
 - (b) acrescentar u elemento no fim de uma lista encadeada;
 - (c) imprimir os conteúdos de todos os elementos de uma lista encadeada.
26. Escreva um TAD PilhaLE que implementa uma pilha usando listas encadeadas.
27. Escreva um TAD FilaLE que implementa uma fila usando listas encadeadas.
28. Discuta vantagens e desvantagens de vetores em relação a listas encadeadas (implementadas num vetor ou dinamicamente). Dê atenção especial as questões de quantidade de memória, velocidade de inserção, remoção e acesso.
29. Dada uma lista encadeada que armazena números inteiros escreva uma função que transforma a lista dada em duas listas encadeadas: a primeira contendo os elementos cujo conteúdo é par e a segunda contendo os elementos com conteúdos ímpares. Sua função deve manipular somente os apontadores e **não** o conteúdo das células. Exemplo:



30. Considere a seguinte situação: um congresso é formado por 400 deputados. Cada deputado está filiado a, no máximo, um partido. Mudanças de partido são permitidas e freqüentes. O presidente do congresso deseja fornecer periodicamente à imprensa listagens com a bancada de cada partido.
- (a) Sugira uma estrutura de dados conveniente para o problema;
 - (b) Faça um procedimento que recebe o nome de um deputado e altera o partido do deputado (dados o partido antigo e o novo);
 - (c) Analise a complexidade do seu procedimento.
31. Uma empresa tem um cadastro que relaciona o número de cada funcionário, seu nível salarial, e seu departamento. O cadastro deve ser organizado de forma que a seguinte operação possa ser feita eficientemente: dado: um departamento; em obter: a relação dos funcionários do departamento em ordem crescente de números. Como as mudanças de funcionários entre departamentos são freqüentes, não é conveniente manter o cadastro ordenado por departamento. O cadastro poderia, então, ser organizado assim:

Matriz A

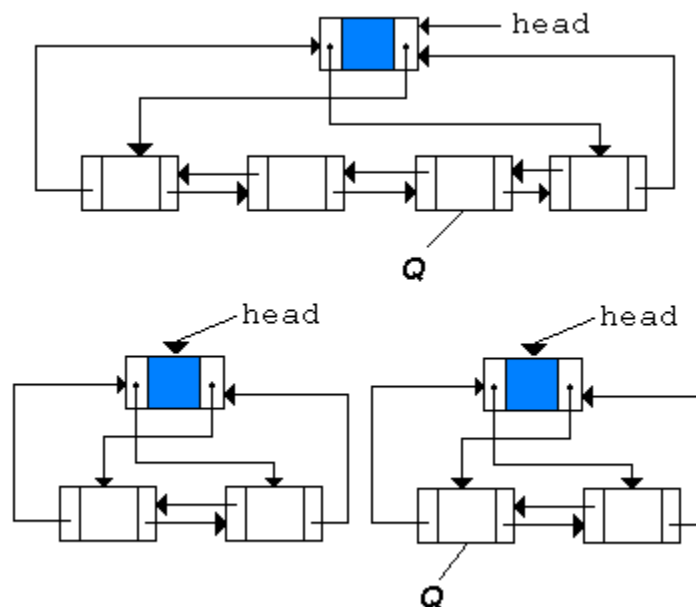
Linha	Funcionário	Nível	Depto	Prox.
1	123	7	1	6
2	8765	12	1	0
3	9210	4	2	0
4	2628	4	3	7
5	5571	8	2	3
6	652	1	1	10
7	7943	1	3	0
8	671	5	3	13
9	1956	11	2	12
10	1398	6	1	11
11	3356	3	1	2
12	4050	2	2	5
13	2468	9	3	4

Matriz B

Depto	Início
1	1
2	9
3	8

A quarta coluna da matriz A estabelece a lista dos funcionários de cada departamento, em ordem crescente de números. A matriz B indica, para cada departamento, a linha do primeiro funcionário do departamento. Escreva procedimentos para executar as seguintes operações sobre as matrizes A e B: (1) admissão de funcionário novo; (2) demissão de funcionário; (3) mudança de departamento por parte do funcionário; (4) relação dos funcionários de um dado departamento.

32. Escreva um método Quebra do TAD lista duplamente encadeada circular, que recebe uma célula Q da lista e devolve uma lista duplamente encadeada circular com os elementos a partir de Q, como mostrado na figura abaixo.



33. Faça o mesmo exercício (32) considerando uma lista circular encadeada.
34. Escreva um algoritmo que recebe uma lista duplamente encadeada e inverte essa lista, alterando somente os campos dos ponteiros e sem usar estruturas auxiliares.