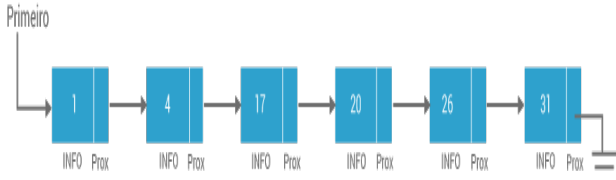




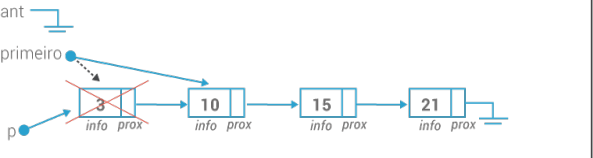
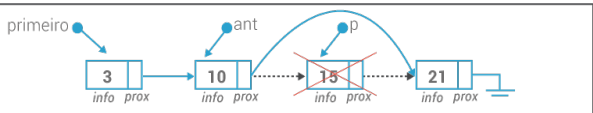


LISTAS ENCADEADAS	LISTAS ENCADEADAS
Como podemos definir uma Lista Encadeada ?	Uma Lista Encadeada é diferente de uma Lista comum (como vetores e matrizes), nas listas comuns é reservado todo um espaço de memória que divide suas alocações com todos valores da lista, se todo o espaço for alocado a lista não poderá armazenar mais valores. Já uma Lista Encadeada não possui um tamanho fixo, o espaço total de memória dela é proporcional ao número de elementos que ela for armazenando. Para isso as Listas Encadeadas não ocupam espaços de memória de forma contígua, ela vai armazenando valores de forma dinâmica em qualquer ponto da memória. Isso faz com que, para percorrer todos os elementos de uma Lista Encadeada tenhamos que guardar a informação do seu endereço de memória junto ao conteúdo armazenado para aquele endereço da lista. Podemos fazer isso por através de ponteiros lógicos.
- reservada para a questão acima -	
Explique como os valores são armazenados numa Lista Encadeada dinamicamente...	A estrutura consiste numa seqüência encadeada de elementos, em geral chamados de nós da lista. A lista é representada por um ponteiro para o primeiro elemento (ou nó). Essa variável (também chamada de cabeça da lista) possibilita o acesso aos demais elementos contidos nela. Do primeiro elemento, pode-se alcançar o segundo seguindo o encadeamento, e assim por diante. O último elemento da lista aponta para NULL, sinalizando que não existe um próximo elemento.
- reservada para a questão acima -	Nesse caso, o primeiro elemento da lista, representado pelo campo info, é 1, o último elemento é 31, o predecessor de 31 é 26, o sucessor de 1 é 4 e assim por diante até o último elemento. Lembrem-se, os elementos dessa lista não estão organizados na memória seqüencialmente, igual a um arranjo. O campo Prox representa o ponteiro para o próximo elemento (endereço onde está armazenado o elemento seguinte da lista). A Lista Encadeada é representada pela Figura abaixo....
- reservada para a questão acima -	
Como seria a estrutura de um nó de uma Lista Encadeada Simples?	Uma lista encadeada (lista ligada) é uma sequência de nós, no qual cada nó contém um objeto de algum tipo e o endereço do nó seguinte. Se os objetos armazenados nos nós são do tipo int, a estrutura de cada nó de uma lista pode ser definida como segue abaixo e é ilustrada pela Figura a seguir.
	<p>Estrutura de um nó</p> 
Como podemos gerar uma Lista Encadeada?	Podemos gerar uma lista Encadeada da seguinte forma:
- reservada para a questão acima -	<pre>struct lista{ int info; struct lista *prox; };</pre>
	Perceba que criamos uma Estrutura que recebe uma variável do tipo inteiro chamada "info", será ela que vai receber os valores dos nós, e dentro da estrutura temos outra estrutura que recebe um endereço de memória chamada prox. Essa estrutura simples criada acima será responsável por criar os nossos nós. Depois podemos atribuir a lista a uma variável chamada "tipo_lista", quando quisermos criar novos nós, basta referenciar a esta variável.
	<pre>typedef struct lista *tipo_lista;</pre>
Como podemos inserir valores na nossa Lista Encadeada?	Podemos fazer isso criando um método de inserção, da seguinte forma:
- reservada para a questão acima -	<pre>void insere_inicio (tipo_lista * primeiro, int elem) { tipo_lista no = (tipo_lista) malloc (sizeof(lista)); no->info=elem; no->prox= *primeiro; *primeiro = no; }</pre>
	Onde, depois de termos a estrutura da nossa lista já criada - como vimos na pergunta anterior - criamos um método que recebe 2 parâmetros: um ponteiro que servirá como ponteiro para o próximo endereço de alocação da lista e outro para uma variável inteira que guardara um valor. Dentro do método criamos uma variável "nó" para o endereço da lista, essa variável vai receber o próprio endereço da lista e irá juntá-lo a memória total da lista por através da função "malloc"; Depois, o "nó" apontando para a variável "info" recebe o valor do elemento enquanto "nó" apontando para a variável "prox" recebe o poteiro passado como parâmetro na função de incremento,m e no fim o valor do ponteiro é atualizado para receber outro nó.

LISTAS ENCADEADAS	LISTAS ENCADEADAS
Ilustre a operação de inserção feita por através do método acima...	<p>Lista vazia</p> 
- reservada para a questão acima -	<p>Inserindo o elemento 1 na Lista</p> 
Como podemos percorrer os elementos de uma Lista Encadeada e mostrá-la na tela?	<p>Inserindo o elemento 5 na Lista</p> 
- reservada para a questão acima -	<p>Para ilustrar a implementação de uma função que percorre todos os elementos da lista, considere a criação de um procedimento que imprima os valores dos elementos armazenados numa lista. Uma possível implementação dessa função é mostrada a seguir.</p> <pre>void imprime_lista (tipo_lista primeiro) { tipo_lista aux; printf ("\nA lista : \n"); aux = primeiro; while (aux!=NULL) { printf("info=%d\t", aux->info); aux=aux->prox; } printf("\n"); }</pre>
Como podemos encontrar um elemento específico em uma Lista Encadeada e mostrá-lo na tela?	<p>Para buscar um elemento na lista é feito o percurso desde o início (primeiro) até que o elemento seja encontrado ou não. Caso o elemento seja encontrado, a busca retorna 1, caso não seja, retorna 0. Uma possível implementação dessa função é mostrada a seguir.</p> <pre>int busca_lista (tipo_lista primeiro, int elem) { tipo_lista aux; aux = primeiro; while (aux!=NULL) { if (aux->info==elem) return 1; aux=aux->prox; } return 0; }</pre>
- reservada para a questão acima -	<p>A função tem como parâmetros de entrada o ponteiro para o primeiro da lista e o valor do elemento que se quer remover. Se o elemento a ser removido for o primeiro da lista, a variável deve ser atualizada. O procedimento para retirar um elemento da lista é mais complexo. Se for descoberto que o elemento a ser retirado é o primeiro da lista, deve-se fazer com que o novo valor da lista passe a ser o ponteiro para o segundo elemento, e então se pode liberar o espaço alocado para o elemento que se quer retirar. Se o elemento a ser removido estiver no meio da lista, se deve fazer com que o elemento anterior a ele passe a apontar para o elemento seguinte, e então se pode liberar o elemento que se quer retirar. Deve-se notar que, no segundo caso, é preciso do ponteiro para o elemento anterior para se poder arrumar o encadeamento da lista. As figuras a seguir ilustram as operações de remoção....</p>
- reservada para a questão acima -	<p>Remove o 1º elemento da Lista</p> 
- reservada para a questão acima -	<p>Remove o 3º elemento da Lista (remoção no meio)</p> 
- reservada para a questão acima -	<p>Remove o último elemento da Lista</p> 