

Prática 5 – Listas Simplesmente Encadeadas

1. Considerando as estruturas de uma lista dinâmica simplesmente encadeada, implemente as seguintes funções em C que operam sobre uma lista. Construa também uma função *main* que faça uso das funções, usando um menu de escolhas. Os parâmetros podem ser ponteiros, ou não, dependendo do caso.
 - a. *FLVazia*
parâmetros: TipoLista *L;
pós-condição: Lista L vazia, com um nó-cabeça;
funcionalidade: cria uma lista vazia com nó-cabeça;
resultado: retorna uma lista vazia criada com nó-cabeça
 - b. *Vazia*
parâmetros: TipoLista L, int flag;
funcionalidade: Testa se a lista está vazia ou não
resultado: flag=1 se a lista está vazia; senão flag=0
 - c. *Retira*
parâmetros: TipoLista *L, TipoItem x, Apontador p;
pré-condição: Lista L tem $n > 1$ elementos, p não é NULL e $p \rightarrow \text{prox}$ não é NULL;
pós-condição: Lista L tem $n - 1$ elementos;
funcionalidade: Retorna o item x que está na posição seguinte ao p da lista mantendo a ordenação dos demais
resultado: elimina o item seguinte ao apontado pelo ponteiro p
 - d. *Inserere*
parâmetros: TipoItem x, TipoLista *L;
pré-condição: Lista L tem $n \geq 0$ elementos;
pós-condição: Lista L tem $n + 1$ elementos;
funcionalidade: Inserere o elemento x após o último elemento da lista
resultado: item inserido após o último item já existente ou nó-cabeça, se lista está vazia
 - e. *Imprime*
parâmetros: TipoLista L;
funcionalidade: Imprime os itens da lista, juntamente com os endereços na memória de cada item (endereço das structs TipoItem)
resultado: impressão dos itens da lista na ordem de ocorrência
2. Os sistemas operacionais e outros aplicativos normalmente registram em arquivo as atividades relevantes dos usuários. Esta relevância é especificada pelo administrador. Por exemplo, um arquivo do tipo *log*, em formato texto, pode ter registrado todos os nomes de usuários que efetuaram *login* em um período, a data do *login*, o horário do *login* e o horário do *logout*. Este arquivo pode ser examinado em busca de atividades suspeitas. Escreva um programa para auxiliar o administrador nesta tarefa.
 - a. Considere a existência de um arquivo fictício *login.log*, onde estão armazenados, em cada linha, nome (*login*) do usuário, data de *login*, horário de *login* e horário de *logout*.
 - b. Construir as estruturas de dados apropriadas para conter os dados.
 - c. Ler o arquivo e preencher uma lista dinâmica simplesmente encadeada.
 - d. Implementar as seguintes funções para o usuário, que serão mostradas em um menu:



- i. *mostraUsuarios*: visualizar na tela todos os nomes que aparecem no arquivo, desprezando repetições de nomes de *login*.
 - ii. *mostraAposHora*: visualizar na tela todos os nomes que efetuaram *login* após um horário indicado pelo administrador.
 - iii. *mostraAcessoUsuario*: visualiza todos os acesso feitos por um determinado nome indicado pelo administrador.
3. Uma empresa precisa organizar os dados de seus funcionários. Estes dados são: nome, matrícula (numérico), departamento, salário. Construa um programa em C e estruturas apropriadas para organizar estes dados em uma lista dinâmica simplesmente encadeada. Use as funções já implementadas no item anterior como auxiliares, modificando-as conforme necessário.