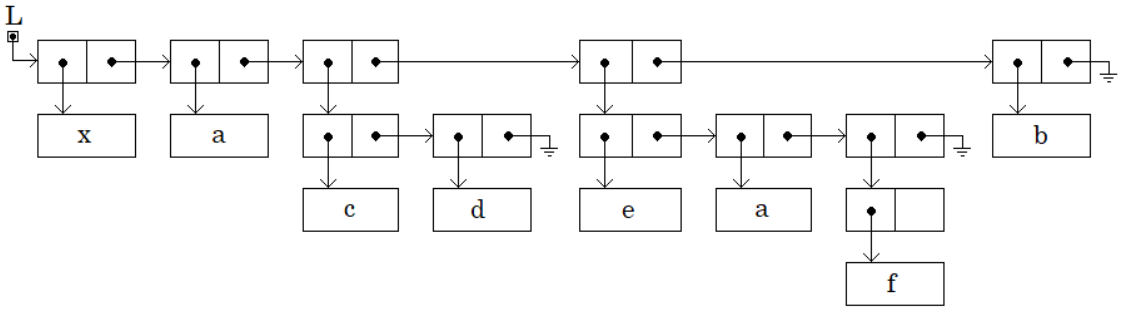
UNOESTE – Universidade do Oeste Paulista
F.I.P.P. – Faculdade de Informática de Presidente Prudente

Estruturas de Dados II

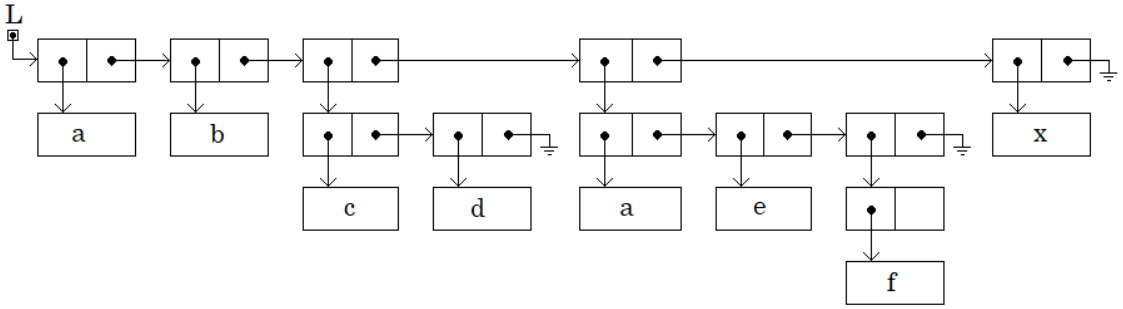
Exercícios de Fixação
(Listas Generalizadas)

1:-) Faça um algoritmo que receba uma Lista Generalizada L por parâmetro e a retorne ordenando todos os átomos de cada uma de suas sublistas. O algoritmo deve ordenar os átomos em cada Lista Generalizada (sublista).
Exemplo:

Lista L:

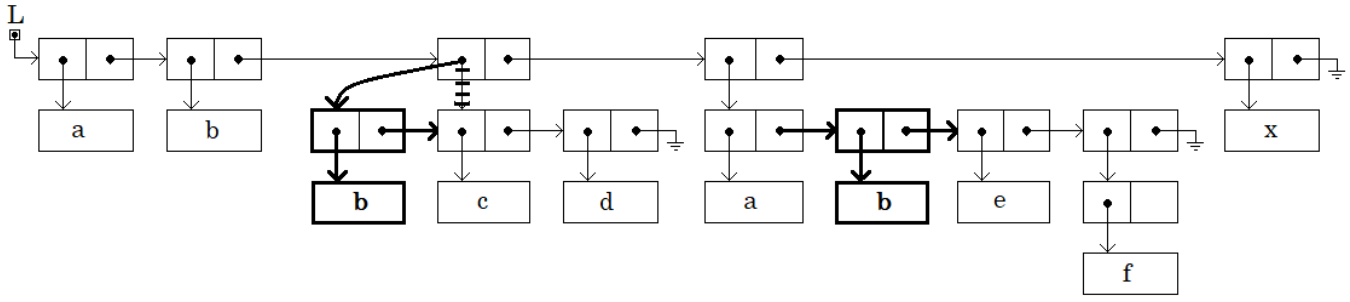


Lista L após ordenação:

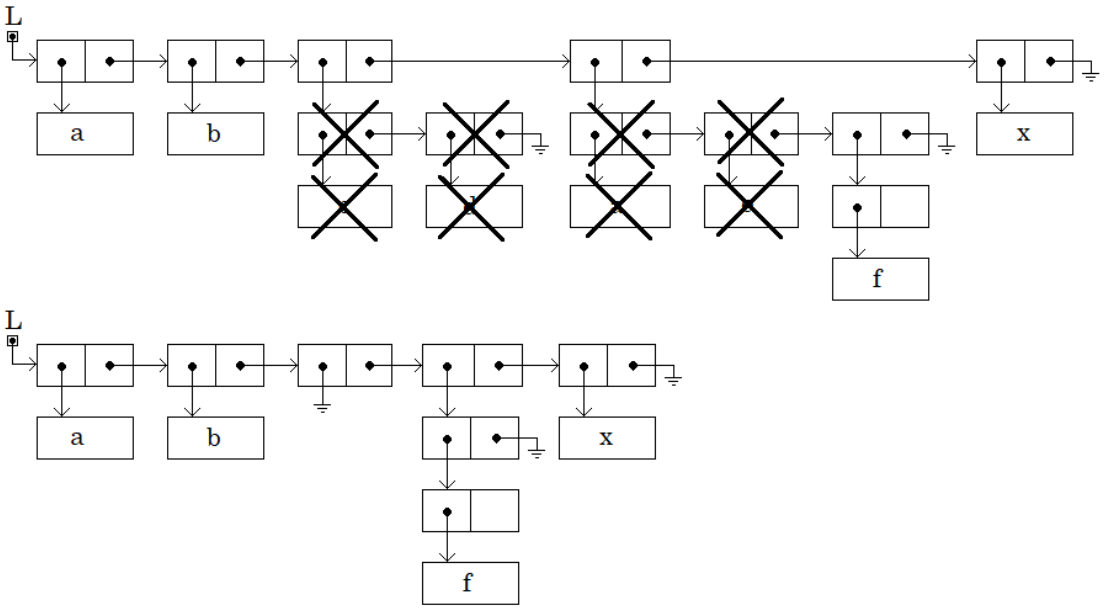


2:-) Faça um algoritmo para inserir um elemento (átomo) em cada uma das sublistas de uma Lista Generalizadas L. A Lista Generalizada possui os átomos em ordem e o elemento deve ser inserido obedecendo essa ordem.
Exemplos de chamada do algoritmo: insere(&L, "b", 1); //Não insere, pois já tem o "b"!
insere(&L, "b", 2);

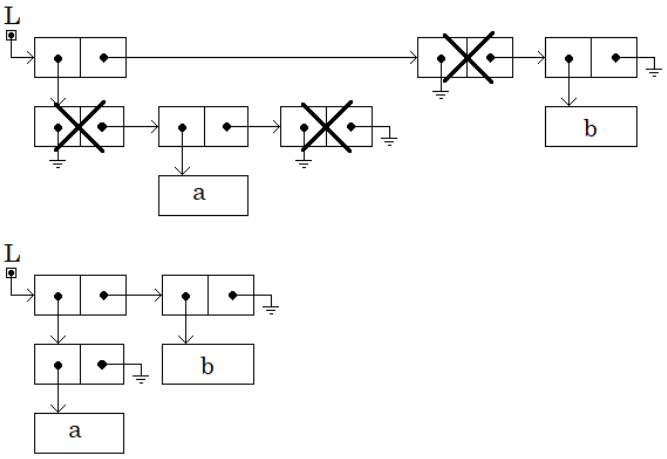
Lista L após a inserção dos elementos:



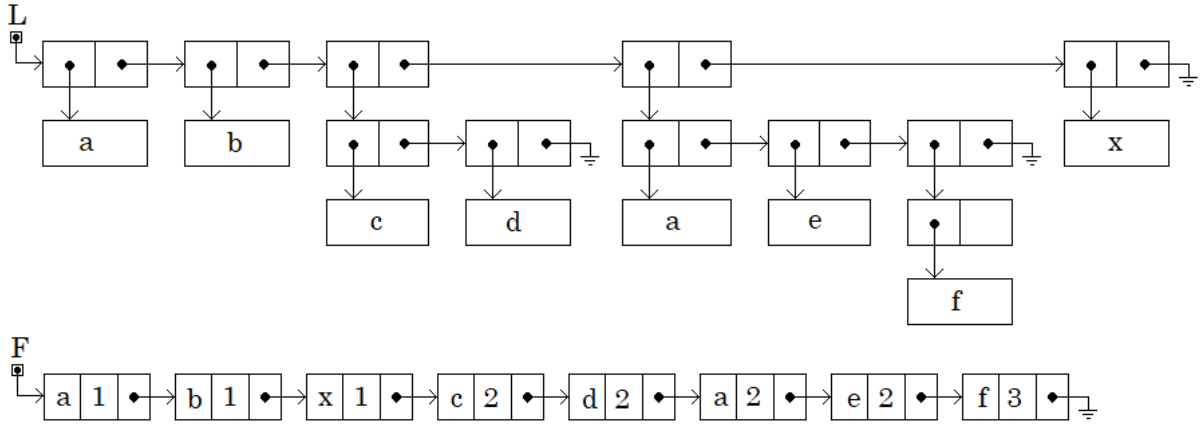
3:-) Faça um algoritmo para excluir os átomos de um dado nível de Lista. Observe que os nodos de Lista que apontam para os átomos também são excluídos!
Exemplos de chamada do algoritmo: exclui_nivel(&L, 2);



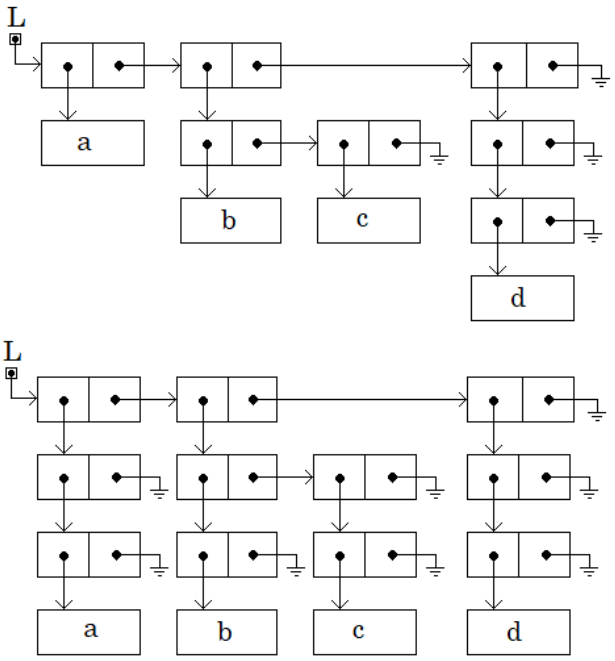
4:-) Faça um algoritmo que exclua todos os nodos de Lista que apontam para Lista Nula.
Exemplo:



5:-) Faça um algoritmo que transforme uma Lista Generalizada em uma Fila com prioridade, sendo que cada nível dos nodos de Lista corresponde a um nível de prioridade.



6:-) Faça um algoritmo que coloque todos os átomos de uma Lista Generalizada L no bottom-level.
Exemplo:



7:-) Faça um algoritmo utilizando Listas Generalizadas para manipular categorias de filmes, sendo que cada categoria pode ser composta de nenhuma, uma ou mais subcategorias. As funções a serem implementadas são:

`inserirCategoria(ListaGen* pai, ListaGen* filho)`
Insere a categoria **filho** como filha da categoria **pai.**

`retirarCategoria(ListaGen* pai, char* nome);`
Retira uma categoria procurando seu nome através da busca a partir de seu **pai**.

`exibirCategorias(ListaGen* pai)`
Exibe a lista de categorias e subcategorias da seguinte maneira (deverá incluir a tabulação):

Comédia
    Policial
    Romântica
Terror
    Zumbi
    Vampiro
Ação
    Aventura
        Anime
        Militar
Ficção Científica
    Poderes
    Invasão espacial

`procurarCategoria(ListaGen* pai, char* nome)`
Retorna a Lista Generalizada filha que se encaixa no parâmetro nome passado.

Exemplo da estrutura:
L = [“Comédia”,[“Policial”,”Romântica”]],[“Terror”,[“Zumbi”,Vampiro”]],[“Ação”,[“Aventura”, [“Anime”, “Militar”]]], [“Ficção Científica”,[“Poderes”, “Invasão espacial”]]

8:-) Considere um polinômio da forma c1*x^n + c2*x^(n-1) + ... + c(n-2)*x^2 + c(n-1)*x + cn*x. Os coeficientes podem ser tanto números reais como outros polinômios. Podemos implementar uma representação desse tipo em C usando union e listas generalizadas. Para isso, uma union deve ser utilizada para guadar um uma estrutura ponteiros para uma lista generalizada que contém a letra que representa a variável (x, y, z, etc.) e uma referência para os coeficientes, ou para um polinômio que contém um valor que representa sua potência e também um valor que será seu coeficiente, que será um valor real ou outra lista. Note, portanto, que para a implementação desse exercício será necessário a utilização de duas unions: uma para representar uma estrutura do polinômio e outra para representar um coeficiente, que guarda um valor real ou uma lista.
Considerando essa estrutura, faça:

a) crie uma função , void exibePolinomio (ListaGen *L), que recebe uma lista generalizada L que representa um polinômio da forma descrita e o exiba da seguinte forma:

(4*x^3 + (2*x)y^2)*z^2 + (-8*-z^4 + 3*x^5)*z^4

b) crie uma função, void avaliaPolinomio(ListaGen *L, float x, float y, float z), que recebe uma lista generalizada L (considere que uma lista só terá as variáveis x, y e z, para fins de simplificação), que representa um polinômio, e retorne seu valor.
Exemplo:
P = (2x^2 + z^3)*y^2, onde x = 2, y = 5 e z = 3. A função deverá retornar 875.