ImprimeTodos(Lista \*L, ){

char X;

boolean temElemento;

PegaoPrimeiro(L, X, temElemento);

while(temElemento == true){

Imprime(X);

PegaoProximo(L,X,temElemento);

}

}

6.2

Interseccao(Lista \*L1, Lista \*L2, Lista \*L3){

char X;

boolean temElemento, ok;

Cria(L3);

PegaoPrimeiro(L1, X, temElemento);

while(temElemento == true){

if(EstaNaLista(L2, X){

Insere(L3, X, ok);

}

PegaoProximo(L2, X, temElemento);

}

}

1 - Procuramos na lista L o nó que guarda o valor X

2 - O eliminamos

Passo 2:

Retira(){ Lista \*L, char X, boolean Ok){

p, anterior;

if(anterior != null) {

anterior->Next = p->Next;

DeleteNode(p);

p = null; anterior = null;

}

}

// caso 2 && 2’

if(anterior == null && L == X) {

L = L-> Next;

DeleteNode(p);

p = null;

}

ok = true;

// caso 3, 4, e 5

ok = false;

}

é necessário procurar e retirar um elemento específico

char ProcuraX(L,X,P,Anterior, AchouX ){

Se x encontrado, p = x nó e anterior pro anterior,

se x primeiro da lista, anterior = null

P = L;

Anterior = null;

while(P != null && P->Info < X){

Anterior = P;

P = P->Next; }

}

if(P != null && P->Info == X){

AchouX = true;

else{AchouX = false;}

}

**Insere(Lista L, char X, boolean ok) {**

Anterior, P = NodePtr;

P = L;

if(Vazia( ))

printf(“Lista vazia, nem vou procurar”);

else {

ProcuraX(L,X,P,Anterior,ok);

if(ok == true)

print(“Ja ta la, n fiz nada”);

else{

while(P->Info < X && P != null){

Anterior = P;

P = P->Next;

}

aux = newNode;

Anterior->Next = aux;

aux->Info = x;

aux->Next = P;

// Anterior->Info = X;

aux = null; P = null; Anterior = null;

}

}

1. Lista Vazia
2. Lista Cheia
3. Inserir no meio
4. inserir no fim
5. inserir no inicio
6. X ja existe//

}

// L primeiro

// L Atual

PegaOPrimeiro(){

}

PegaOProximo(){

}