|  |
| --- |
| ***ALGORITMO .::. inicializarLDEArr*** |
| *Entrada*: LD (Tipo Lista Duplamente Encadeada Estática)  IA, FA (inteiros)  *Saída*: LD (Tipo Lista Duplamente Encadeada Estática)  IA, FA, IL, FL, LD.Disp (inteiros) |
| **Início**  LD.Disp←1;  **Início**  **Para** Ind de IA incr 1 **até** FA **faça**  LD[ind].dir ← Ind+1  LD[ind].esq ← Ind-1  **Fim**  LD[IA].dir ← -1  LD[IA].esq ← -1  LD[FA].dir ← -1  **Fim** |
|  |

|  |
| --- |
| ***ALGORITMO .::. alocarNoLDArr*** |
| *Entrada*: LD (Tipo Lista Duplamente Encadeada Estática)  IA, FA, IL, FL, LD.Disp (inteiro)  *Saída*: LD (Tipo Lista Duplamente Encadeada Estática)  IA, FA, IL, FL, LD.Disp (inteiro)  sucesso (lógico) |
| **início**  ind(inteiro)  **Se** (LD.Disp = -1) **então**  sucesso←falso  **Senão**  **início**  ind←LD.Disp  LD.disp←LD[LD.disp].dir  **Retorna**(ind)  **fim**  **fim** |
|  |

|  |
| --- |
| ***ALGORITMO .::. inserirDepoisLDArr*** |
| *Entrada*: LD (Tipo Lista Duplamente Encadeada Estática)  IA, FA, IL, FL, LD.Disp, Ipo (inteiro)  infNo (Tipo nó)  *Saída*: LD (Tipo Lista Duplamente Encadeada Estática)  IA, FA, IL, FL, LD.Disp (inteiro) |
| **Início** ind,ind2(inteiros)  ind←AlocarNoLDArr(LD)  LD[ind].info ← infNo  ind2 ← LD[Ipo].dir  **Inicio**  **Se** (ind2 = -1) **então** LD[ind2].esq = ind  **Fim**  LD[ind].dir ← ind2  LD[ind].esq ← Ipo  LD[Ipo].dir ← ind  **Fim** |

|  |
| --- |
| ***ALGORITMO .::. liberarNoLDArr*** |
| *Entrada*: LD (Tipo Lista Duplamente Encadeada Estática)  IA, FA, IL, FL,  Ipo (inteiro)  *Saída*: LD (Tipo Lista Duplamente Encadeada Estática)  IL, FL (inteiro) |
| **Início**  LD[Ipo].dir = LD.Disp  LD.Disp ← Ipo  **Fim** |

|  |
| --- |
| ***ALGORITMO .::.removerNoLDArr*** |
| *Entrada*: LD (Tipo Lista Duplamente Encadeada Estática)  IL, FL, IA, FA (inteiro)  Ipo (inteiro)  *Saída*: LD (Tipo Lista Duplamente Encadeada Estática)  IL, FL (inteiro)  sucesso (lógico) |
| **início**  ind,ind2(inteiro)  **Início**  **se** Ipo = 0 **ou** Ipo = -1 **então**  sucesso ← falso  **senão  Inicio**  ind ← LD[Ipo].esq  ind2 ← LD[Ipo].dir  **Inicio**  **se** ind2 = -1 **então**  LD[ind2].esq ← ind  **Fim**  LD[ind].dir ← ind2  liberarNoLDArr(LD,Ipo)  **Fim**  **Fim**  **fim** |