**LISTA LINEAR COM DESCRITOR**

|  |
| --- |
| **ALGORITMO .::. inicializarLLArrDesc** |
| *TipoDescritor*  *registro*  *IA, IL, N, FL, FA: inteiros  fim registro*  *Entrada*: DL (TipoDescritor); iniArea, fimArea (inteiro)  *Saída*: DL (TipoDescritor) |
| **início**  DL.IL←0;  DL.FL←0;  DL.N←0;  DL.IA←iniArea;  DL.FA←fimArea;  **Fim** |

|  |
| --- |
| **ALGORITMO .::. inserirLLArrDesc** |
| *Entrada*: LL (tipoLista); DL (tipoDescritor); infNo (Tipo nó); pos (inteiro)  *Saída*: LL (tipoLista); DL (tipoDescritor); sucesso (lógico) |
| **início**  **Se** (DL.N=DL.FA-DL.IA+1) OU (pos>DL.N+1) OU (pos<1) **então**  sucesso←falso  **Senão**  **inicio**  sucesso←verdadeiro  DL.N←DL.N+1  **Se** (pos=1) **então {**INSERIR NO INICIO**}**  **inicio**  **Se** (DL.N=1) **então**  **inicio**  DL.IL←(DL.FA-DL.IA+1) div 2  DL.FL←DL.IL  **fim**  **senão**  **Se** (DL.IL>DL.IA) **então**  DL.IL←DL.IL-1  **Senão**  **inicio**  **para** K **de** DL.FL **até** DL.IL **incr** -1 **faça**  LL[k+1]←LL[k]  DL.FL←DL.FL+1  **fim**  LL[DL.IL]←info  **fim**  **Senão**  **Se** (pos=DL.N) **então {**INSERIR NO FIM**}**  **inicio**  **Se** (DL.FL<DL.FA) **então**  DL.FL←DL.FL+1  **senão**  **inicio**  **para** K **de** DL.IL **até** DL.FL **faça**  LL[k-1]←LL[k]  DL.IL←DL.IL-1  **fim**  LL[DL.FL]←info  **fim**  **Senão {**INSERIR NO MEIO**}**  **inicio**  **Se** (DL.FL<DL.FA) **então**  **inicio**  indPos←DL.IL+Pos-1  **para** K **de** DL.FL **até** indPos **inc** -1 **faça**  LL[k+1]←LL[k]  DL.FL←DL.FL+1  **fim**  **Senão**  **inicio**  indPos←DL.IL+Pos-2  **para** K **de** DL.IL **até** indPos **inc** +1 **faça**  LL[k-1]←LL[k]  DL.IL←DL.IL-1  **fim**  LL[indPos]←info  **fim**  **fim**  **fim** |

|  |
| --- |
| **ALGORITMO .::. removerKLLArrDesc** |
| *Entrada*: LL (tipoLista) ; DL(tipoDescritor); K (inteiro)  *Saída*: LL (tipoLista) ; DL(tipoDescritor); sucesso (lógico) |
| **início**  **Se** (K≤0)OU(K>DL.N) **então**  sucesso←falso  **Senão**  **inicio**  sucesso←verdadeiro  **para** Ind **de** DL.IL+K-1 **até** DL.FL-1 **inc** +1 **faça**  LL[Ind]←LL[Ind+1]  DL.FL←DL.FL-1  DL.N←DL.N-1  **Se** (DL.FL=DL.IL-1) **então**  DL.IL←DL.FL←0  **fim**  **fim** |

|  |
| --- |
| **ALGORITMO .::. acessarKLLArrDesc** |
| *Entrada*: LL (tipoLista); DL (tipoDescritor); K (inteiro)  *Saída*: infNo (tipoNo); sucesso (lógico) |
| **início**  **Se** (K≤0)OU(K>DL.FL-DL.IL+1)OU(DL.N=0) **então**  sucesso←falso  **Senão**  **inicio**  infNo←LL[DL.IL+K-1]  Sucesso = verdadeiro  **fim**  **fim** |

|  |
| --- |
| **ALGORITMO .::. posValLLArrDesc** |
| *Entrada*: LL (tipoLista); DL (tipoDescritor); valBusca (tipoValor)  *Saída*: posição (inteiro) |
| **início**  achou←falso  posição←0  I←DL.IL  **enquanto** (I≤DL.FL)E(não achou) **faça**  **se** LL[I].valor = valBusca **então**  **inicio**  posição←I  achou←verdadeiro  **fim**  **senão**  I←I+1  **fim** |