Alocação Encadeada - Exercício

Implemente, no TAD LISTA_ENC, utilizando recursividade, a seguinte operação:

void gera_lista (LISTA_ENC *pl,int m,int n)

a qual utilizando-se das operações do TAD LISTA produz uma lista de inteiros correspondente a [m..n].

Alocação Encadeada

Com base em nossos novos conhecimentos adquiridos, podemos agora definir um novo TAD LISTA_ENC_ORD, na qual os elementos encontram-se ordenados de forma crescente ou decrescente, ou seja, no caso da ordenação crescente, o primeiro elemento é menor que o segundo, que por sua vez é menor que o terceiro e assim sucessivamente. (Para exemplificar, consideraremos a ordem crescente)

```
typedef struct nodo
{
   int inf;
   struct nodo * next;
}NODO;
typedef NODO * LISTA_ENC_ORD;
void cria_lista (LISTA_ENC_ORD *);
int eh_vazia (LISTA_ENC_ORD);
int tam (LISTA_ENC_ORD);
void ins (LISTA_ENC_ORD *, int);
int recup (LISTA_ENC_ORD, int);
void ret (LISTA_ENC_ORD *, int);
```

Alocação Encadeada

Com uma pequena análise, percebe-se que a única operação que requer alteração do TAD LISTA_ENC para o TAD LISTA_ENC_ORD é a operação de inserção.

Implementaremos agora esta operação.

```
void ins (LISTA_ENC_ORD *pl, int v)
{
   NODO *novo;
   novo = (NODO *) malloc (sizeof(NODO));
   if (!novo)
   {
      printf ("\nERRO! Memoria
    insuficiente!\n");
      exit (1);
   }
   novo->inf = v;
```

```
if (*pl==NULL || v<(*pl)->inf)
{
   novo->next = *pl;
   *pl = novo;
}
else
{
   NODO *aux;
   for (aux=*pl; aux->next!=NULL &&
 v>(aux->next)->inf;  aux=aux->next);
   novo->next = aux->next;
   aux->next = novo;   } }
```

Alocação Encadeada - Exercício

Implemente, no TAD LISTA_ENC_ORD, a seguinte operação:

void ret_com_base_no_valor (LISTA_ENC_ORD *, int);

a qual recebe uma referência para uma lista e um valor que deve ser retirado desta.