

Seja uma lista encadeada organizada da seguinte maneira:

1. Uma estrutura (tipo **Lista**), que armazena um ponteiro para o primeiro elemento (**prim**) e um ponteiro para o último elemento da lista (**ult**). Caso a lista esteja vazia, ambos os campos (**prim** e **ult**) terão NULL;
2. Zero ou mais nós (tipo **Elemento**), onde cada nó armazena a informação em si (campo **info**) e um ponteiro para o próximo nó da lista.

Essa lista é usada para representar a lista de tarefas a serem executadas por cada funcionário da empresa. Os tipos de dados usados para a definição dessa lista são os seguintes:

```
struct tarefa {
    char urgencia;        // 'S' se urgente, 'N' em caso contrario
    char descricao[81];
    int  matrSolicitante;
};
typedef struct tarefa Tarefa;

struct elemento
{
    Tarefa *info;
    struct elemento *prox;
};
typedef struct elemento Elemento;

struct lista
{
    Elemento *prim;
    Elemento *ult;
};
typedef struct lista Lista;
```

Quando surge uma nova tarefa, ela é incluída no final da lista.

A próxima tarefa a ser executada é sempre a primeira tarefa urgente da lista, ou seja, a primeira tarefa que apareça na lista cuja urgência (**urgencia**) seja igual a 'S'. Caso não haja tarefas urgentes, executa-se a primeira tarefa que apareça na lista.

Escreva em C a função *lst_retira*, que recebe uma lista (*lst*) e retorna um ponteiro para a próxima tarefa a ser executada ou NULL, caso não existam tarefas. O nó que possui a tarefa a ser executada deve ser eliminado da lista. Observe que ao eliminar o nó da lista, pode ter que se atualizar os campos que descrevem a lista nas seguintes situações:

- Se o nó a ser eliminado for o primeiro da lista, o campo **prim** deverá apontar para o nó seguinte ao que será eliminado;
- Se o nó a ser eliminado for o único que existe na lista, o campo **ult** terá que ser atualizado;
- Se o nó a ser eliminado for o último da lista, o campo **ult** deverá apontar para o nó anterior ao que será eliminado.

O protótipo da função *lst_retira* é o seguinte:

```
Tarefa* lst_retira (Lista* lst);
```