

## Melhores momentos

## Listas encadeadas

## AULA 8

### Estrutura de uma lista encadeada em C

```
struct celula {
    int conteudo;
    struct celula *prox;
};
typedef struct celula Celula;

Celula *ini;
/* inicialmente a lista esta vazia */
ini = NULL;
```



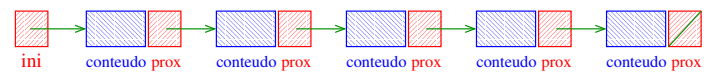
### Busca em uma lista encadeada

Esta função **recebe** um inteiro **x** e uma lista **ini**. A função **devolve** o endereço de uma célula que contém **x**. Se tal célula não existe, a função **devolve** **NULL**.

```
Celula *busca (int x, Celula *ini)
{
    Celula *p;
    p = ini;
    while (p != NULL && p->conteudo != x)
        p = p->prox;
    return p;
}
```

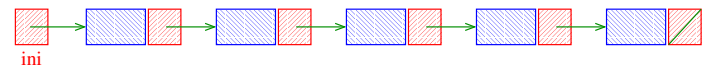
Uma **lista encadeada** (= *linked list* = lista ligada) é uma sequência de **células**; cada **célula** contém um **objeto** de algum tipo e o **endereço** da célula seguinte.

Ilustração de uma **lista encadeada**



### Imprime conteúdo de uma lista

Esta função **imprime** o **conteúdo** de cada célula de uma lista encadeada **ini**.



```
void imprima (Celula *ini)
{
    Celula *p;
    for (p=ini; p != NULL; p=p->prox)
        printf("%d\n", p->conteudo);
    printf("\n");
}
```

## AULA 9

## Mais listas encadeadas em C

PF 4, S 3.3

<http://www.ime.usp.br/~pf/algoritmos/aulas/lista.html>

### Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento  $x$  e insere esta célula no início da lista  $ini$ .

```
void insere (int x, Celula *ini)
{
    Celula nova;    ERRADO!
    nova.conteudo = x;
    nova.prox = ini;
    ini = &nova;
}
```

A estrutura  $nova$  é uma **variável local** (=automática). Ela será devolvida ao sistema no final da execução da função.

### Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento  $x$  e insere esta célula no início da lista  $ini$ .

```
void insere (int x, Celula *ini)
{
    Celula *nova;
    nova = mallocSafe(sizeof(Celula));
    nova->conteudo = x;
    nova->prox = ini;
    ini = nova;    ERRADO!
}
```

**Mesmo erro!** O parâmetro  $ini$  é uma **variável local** (=automática) que começa inicializada com valor do argumento.

### Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento  $x$  e insere esta célula no início da lista  $ini$ .

```
Celula *insere (int x, Celula *ini) {
    Celula *nova;
    nova = mallocSafe(sizeof(Celula));
    nova->conteudo = x;
    nova->prox = ini;
    return nova;
}
```

**Certo!** Exemplos de chamadas da função

```
ini = insere(11, ini);
ini = insere(valor+3, ini);
```

### Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento  $x$  e insere esta célula no início da lista  $ini$ .

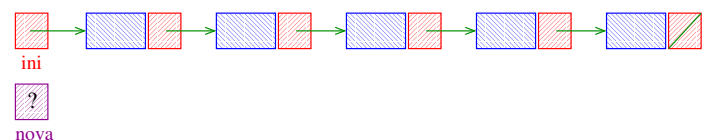
```
void insere (int x, Celula **ini) {
    Celula *nova;
    nova = mallocSafe(sizeof(Celula));
    nova->conteudo = x;
    nova->prox = *ini;
    *ini = nova;
}
```

**Certo!** Exemplos de chamadas da função

```
insere(11, &ini);
insere(valor+3, &ini);
```

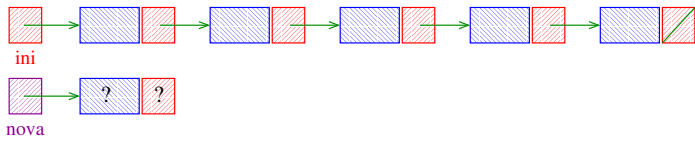
### Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento  $x$  e insere esta célula no início da lista  $ini$ .



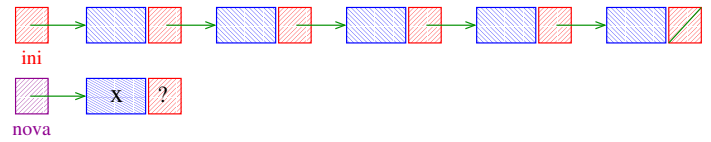
## Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento  $x$  e insere esta célula no início da lista  $ini$ .



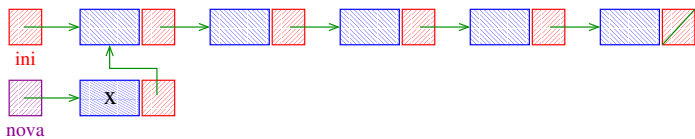
## Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento  $x$  e insere esta célula no início da lista  $ini$ .



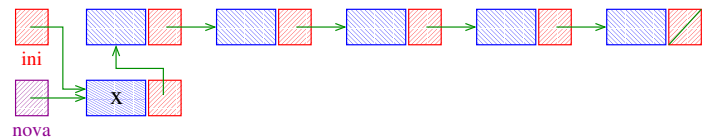
## Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento  $x$  e insere esta célula no início da lista  $ini$ .



## Inserção no início de uma lista encadeada

Cria uma célula para guardar um elemento  $x$  e insere esta célula no início da lista  $ini$ .



## Remoção em uma lista encadeada

Recebe o endereço  $p$  de uma célula de uma lista encadeada e remove da lista a célula  $p \rightarrow prox$ .

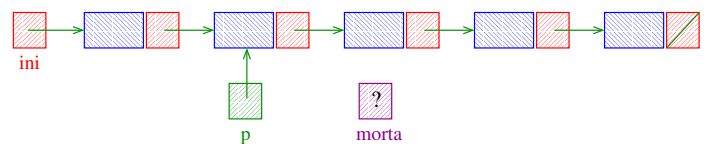
A função supõe que  $p \neq \text{NULL}$  e  $p \rightarrow prox \neq \text{NULL}$ .

```
void remove (Celula *p)
{
    Celula *morta;
    morta = p->prox;
    p->prox = morta->prox;
    free(morta);
}
```

## Remoção em uma lista encadeada

Recebe o endereço  $p$  de uma célula de uma lista encadeada e remove da lista a célula  $p \rightarrow prox$ .

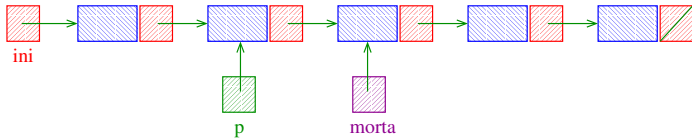
A função supõe que  $p \neq \text{NULL}$  e  $p \rightarrow prox \neq \text{NULL}$ .



## Remoção em uma lista encadeada

Recebe o endereço  $p$  de uma célula de uma lista encadeada e  $remove$  da lista a célula  $p \rightarrow prox$ .

A função supõe que  $p \neq \text{NULL}$  e  $p \rightarrow prox \neq \text{NULL}$ .

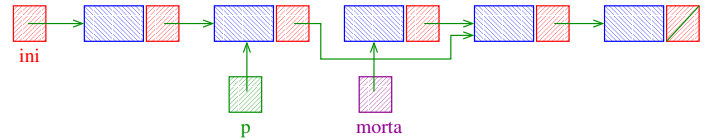


Navigation icons: back, forward, search, etc.

## Remoção em uma lista encadeada

Recebe o endereço  $p$  de uma célula de uma lista encadeada e  $remove$  da lista a célula  $p \rightarrow prox$ .

A função supõe que  $p \neq \text{NULL}$  e  $p \rightarrow prox \neq \text{NULL}$ .

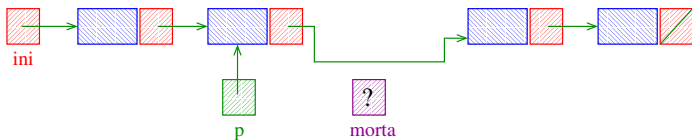


Navigation icons: back, forward, search, etc.

## Remoção em uma lista encadeada

Recebe o endereço  $p$  de uma célula de uma lista encadeada e  $remove$  da lista a célula  $p \rightarrow prox$ .

A função supõe que  $p \neq \text{NULL}$  e  $p \rightarrow prox \neq \text{NULL}$ .



Navigation icons: back, forward, search, etc.

## Busca e Inserção em uma lista

Recebe uma lista  $ini$  e  $insere$  uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , insere a célula com  $y$  no final da lista.

Celula \*

```
buscaInsere(int x, int y, Celula *ini) {
    Celula *p, *q, *nova;
    nova = mallocSafe(sizeof(Celula));
    nova->conteudo = x;
    if (ini == NULL || ini->conteudo == y){
        nova->prox = ini;
        ini = nova;
    }
}
```

Navigation icons: back, forward, search, etc.

## Busca e Inserção em uma lista

```
else {
    p = ini;
    q = p->prox;
    while (q != NULL && q->conteudo != y){
        p = q;
        q = p->prox;
    }
    p->prox = nova;
    nova->prox = q;
}
return ini;
}
```

Navigation icons: back, forward, search, etc.

## Chamadas de buscaInsere

Celula \*ini, \*ini2;  
ini = ini2 = NULL;

[...manipulação das listas ...]

```
ini = buscaInsere(22, 33, ini);
ini2 = buscaInsere(x+1, y, ini2);
ini2 = buscaInsere(x, 2*y, ini2);
ini = buscaInsere(valor, meio, ini);
```

Navigation icons: back, forward, search, etc.

## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



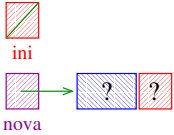
## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



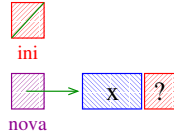
## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



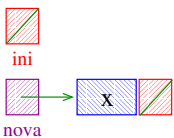
## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



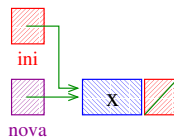
## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



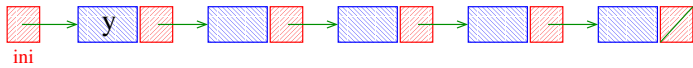
## Busca e Inserção em uma lista

Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



## Busca e Inserção em uma lista

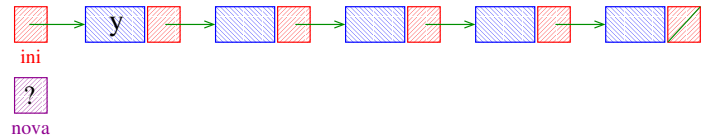
Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

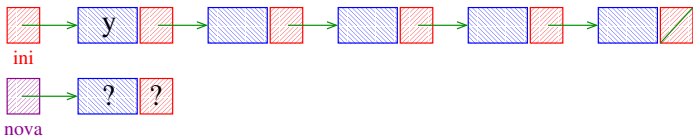
Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

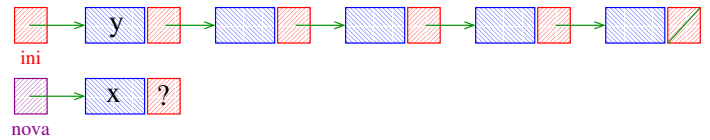
Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

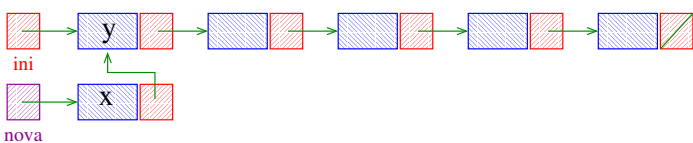
Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

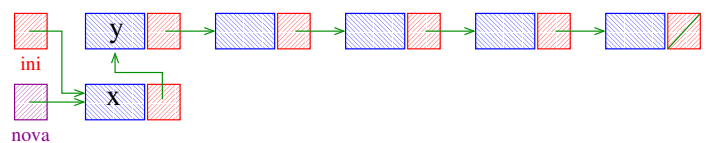
Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

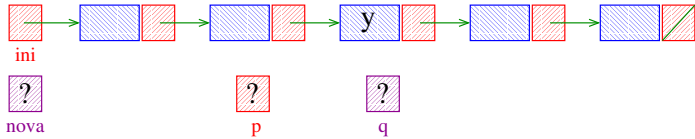
Inserir uma célula de conteúdo  $x$  antes da primeira célula de conteúdo  $y$ . Se nenhuma célula contém  $y$ , inserir a célula com  $y$  no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

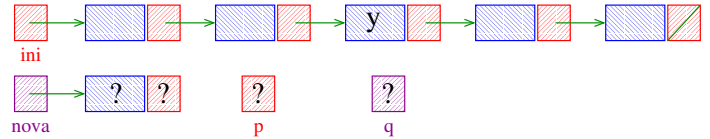
Inserir uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, inserir a célula com **y** no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

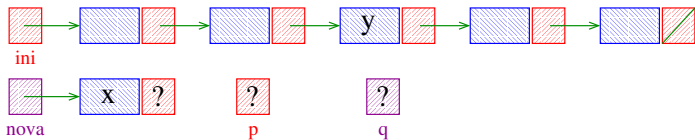
Inserir uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, inserir a célula com **y** no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

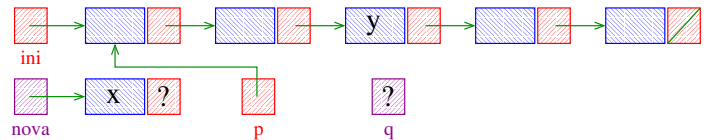
Inserir uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, inserir a célula com **y** no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

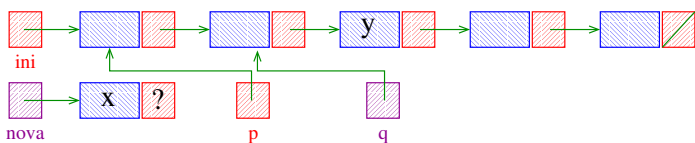
Inserir uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, inserir a célula com **y** no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

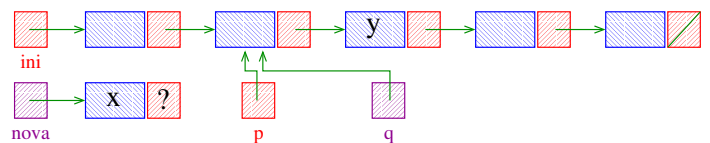
Inserir uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, inserir a célula com **y** no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

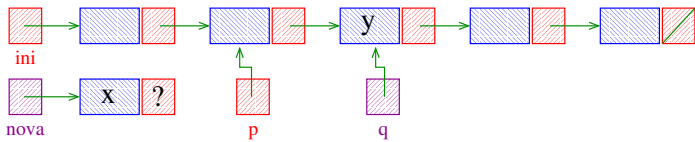
Inserir uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, inserir a célula com **y** no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

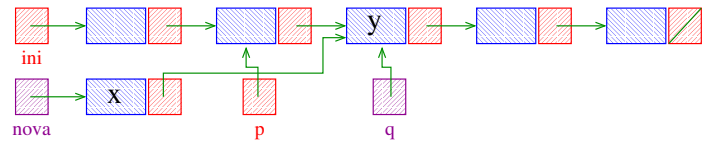
Inserir uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, inserir a célula com **y** no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

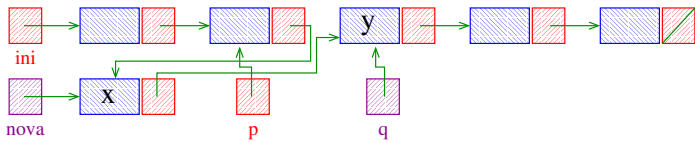
Inserir uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, inserir a célula com **y** no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

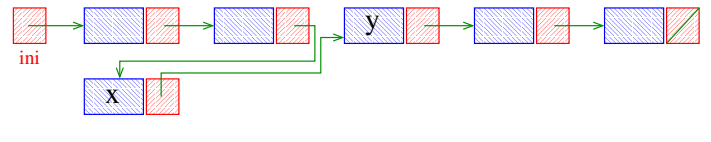
Inserir uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, inserir a célula com **y** no final da lista.



< > < > < > < > < > < >

## Busca e Inserção em uma lista

Inserir uma célula de conteúdo **x** antes da primeira célula de conteúdo **y**. Se nenhuma célula contém **y**, inserir a célula com **y** no final da lista.



< > < > < > < > < > < >