Listas – continuação Exclusão início e meio(antes do k)

Profa. Fabrícia Damando Santos
fabriciadamando@gmail.com

ATÉ AGORA:

- Lista com 1-2 elementos
- Lista com função:
- Inserção no final
- Exclusão no final

PRÓXIMA META:

- Excluir elemento no início ou no meio da lista

REMOÇÃO - EXCLUSÃO

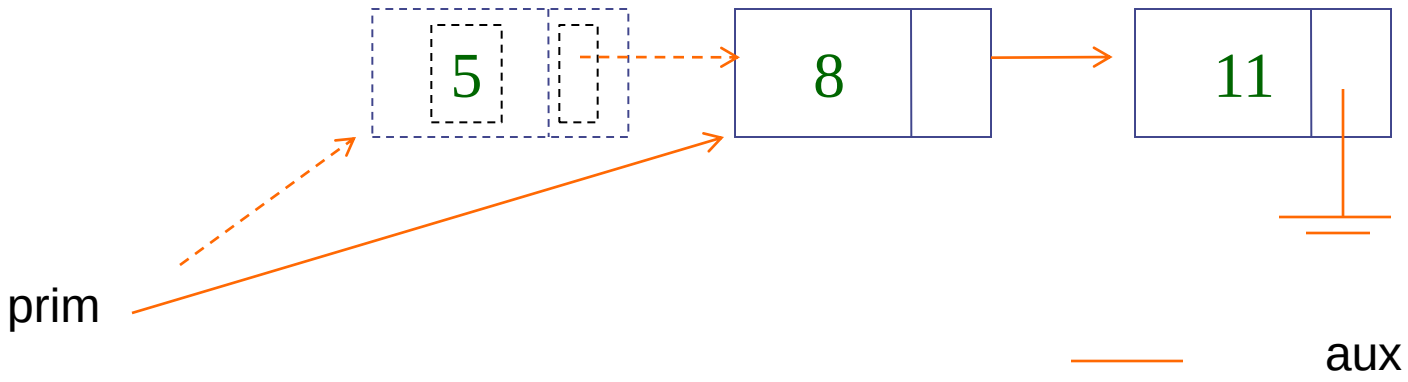
- A remoção permite liberar espaço de memória referente ao nodo especificado
- As remoções – são funções que retornam o valor que está sendo excluído da lista
- A remoção do primeiro nodo – à esquerda
- A remoção do último nodo – à direita
- Libera-se a memória

REMOÇÃO DO PRIMEIRO NODO (À ESQUERDA)

- Remoção do primeiro nodo / remoção à esquerda:
- Libera a memória alocada por esse nodo
- Atualiza p conteúdo da variável início com o endereço do primeiro nodo da lista
- Para remover o primeiro nodo:
  - Verifica se a lista está vazia
    - Início = NULL /// a lista está vazia e não tem como remover
  - Se a lista não está vazia (enquanto p->proximo <> NULL:
    - O conteúdo da variável início – será o endereço de memória do primeiro nodo, <> de NULO
  - Se tiver somente um nodo – p->proximo = NULL
    - Libera memória
    - Início = NULL

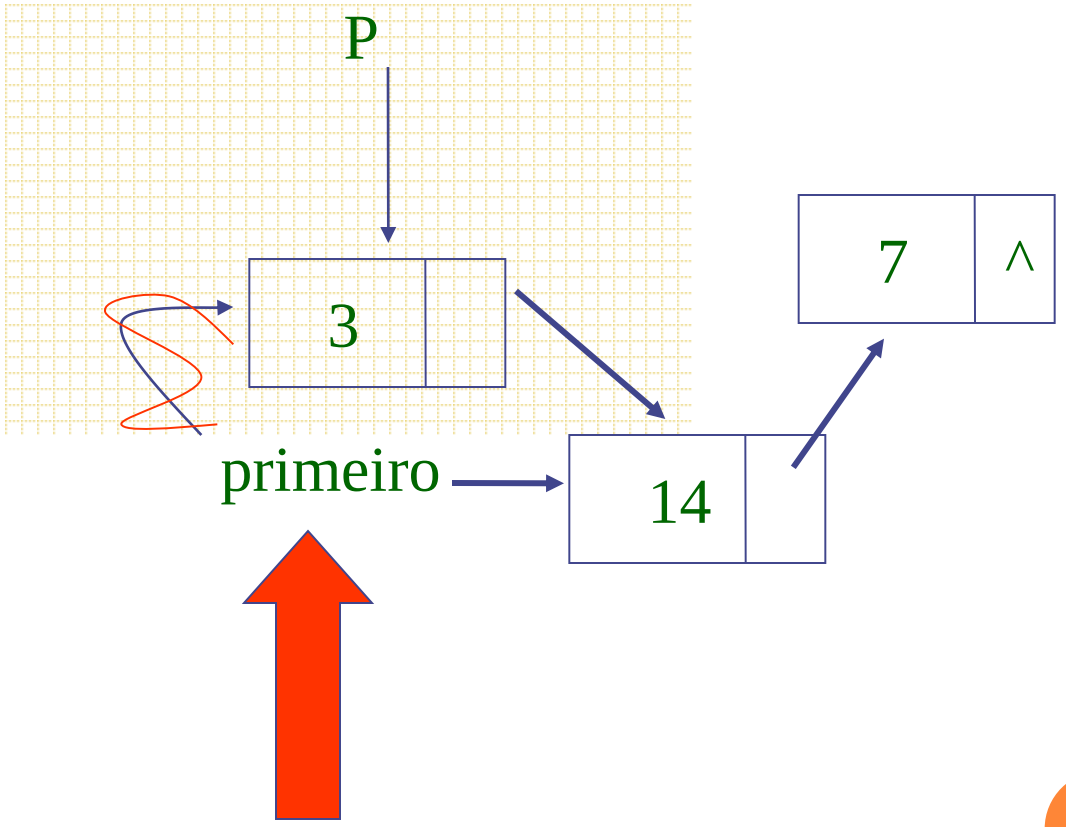
Remoção na lista

- Parâmetros (Entrada da lista, Valor a retirar)
- Se for primeiro nó
  - Devemos fazer com que o novo valor da lista passar a ser o ponteiro para o segundo elemento
  - Libera-se o espaço para o elemento que queremos retirar



- Para remover
  - Passa para a função o ponteiro inicio e fim
  - Declara nodo *pont, *aux
  - pont = *inicio
  - Se o pont->proximo = NULL
    - Não tem mais elementos na lista
    - Libera o ponteiro -Free(pont)
    - Inicio e fim = NULL – A lista está vazia
  - Caso contrário – se tem elementos na lista
    - Aux = pont->prox.imo
    - *inicio = pont->proximo
    - Free(pont)

Lista encadeada – remoção do primeiro nodo



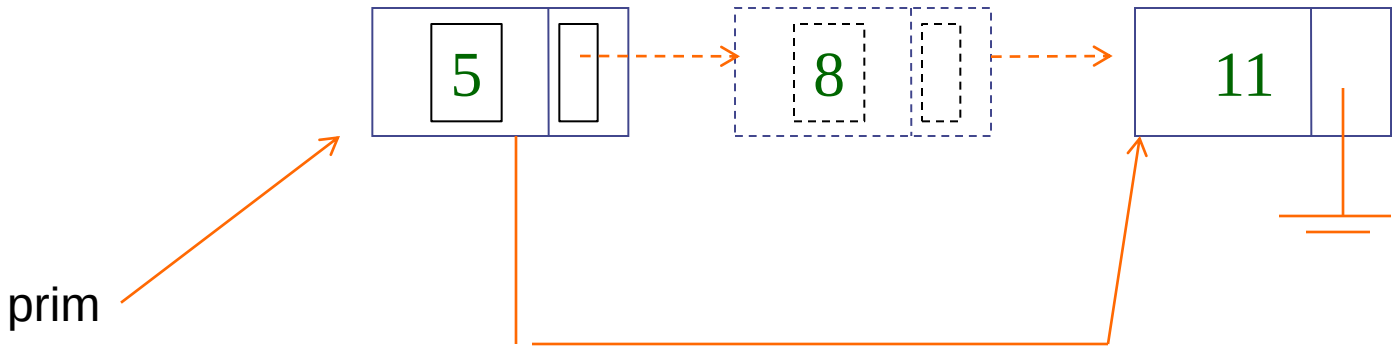
Remoção no meio da lista – antes do K

- Remove o nó antes do nó indicado
- A exclusão no ocorre ser:
  - A lista estiver vazia (inicio = NULL)
  - Se K é o primeiro nodo da lista (inicio = K)
- Se nenhuma das opções acima ocorre, pode-se excluir
  - O nodo K deve ser localizado na lista
  - O nodo anterior deve ser excluído

```
p=inicio
enquanto (p<> nulo e p->proximo <> K) //percorre a lista
ant = p
p = p->proximo
```

Remoção da lista

- Meio da lista
  - O elemento anterior a ele passar a apontar para o seguinte
  - Libera-se aquele que se quer retirar
  - Precisamos de um ponteiro para o elemento anterior para continuar o encadeamento



LISTA ENCADEADA – REMOÇÃO ANTES DO K

```
funcao Remove_antes_k (ref registro nodo *primeiro, *k) : inteiro
inicio
   int valor;
   registro nodo *p;
   se (primeiro = nulo) { Se Lista estiver vazia ...}
   entao valor = 0        { retorna valor 0 ...}
   senao inicio

            se (primeiro = k)  { Se Inicio da lista for igual ao nodo K  ? }
            entao valor = 0  { Erro, pois o nodo K é o primeiro nodo da lista..}
            senao inicio
                     p := primeiro;   { p representa o nodo aux que vamos remover.. }
                     aux := primeiro;   { aux representa o nodo k }
                     enquanto (p->proximo <> K) e (p->proximo <> nulo) { Percorre a lista  }
                     faca inicio
                              aux = p;
                              p = p->proximo
                            fim
                     se (primeiro = p) { Se P for o primeiro nodo da lista...}
                     entao primeiro = k
                     senao aux->proximo = k;
                     valor = p->dados;
                                                                  fim
                                                                  retorna valor;


                                                                fim;
```

Exercício

- Fazer uma lista que possa:
  - Inserir no final
  - Imprimir
  - Excluir do final
  - Exclui no início
  - Excluir no meio

  - Ter um menu de opções para o usuário
  - Imprimir
    - Endereço valor próximo.
  - Verificar se a lista não está vazia
  - Quando excluir todos avisar ao usuário que a lista encontra-se vazia