

Estrutura de Dados



Aula 04 – Listas baseadas em Vetores

Prof. Marcos Nava

Listas



❧ O que é uma lista?



Listas



- ❧ Uma lista é uma coleção de itens
- ❧ Uma lista não precisa necessariamente ser ordenada, mas deve ter um recurso para ordenar suas informações
- ❧ Uma lista deve prover acesso aleatório aos itens

Listas



- ❧ Não existem ordem certa para inserir os dados na lista, nem para retirá-los
- ❧ Quando um dado é inserido, ele deve ser colocado no final da lista
- ❧ Quando um dado é excluído, o espaço utilizado pelo dado deve ser ocupado pelo imediatamente posterior e todos os outros devem ser realocados

Listas



☞ Veja as funções que seriam interessante implementar:

- add
- size
- searchByIndex
- searchByValue
- delete
- sort
- set
- expand

Listas

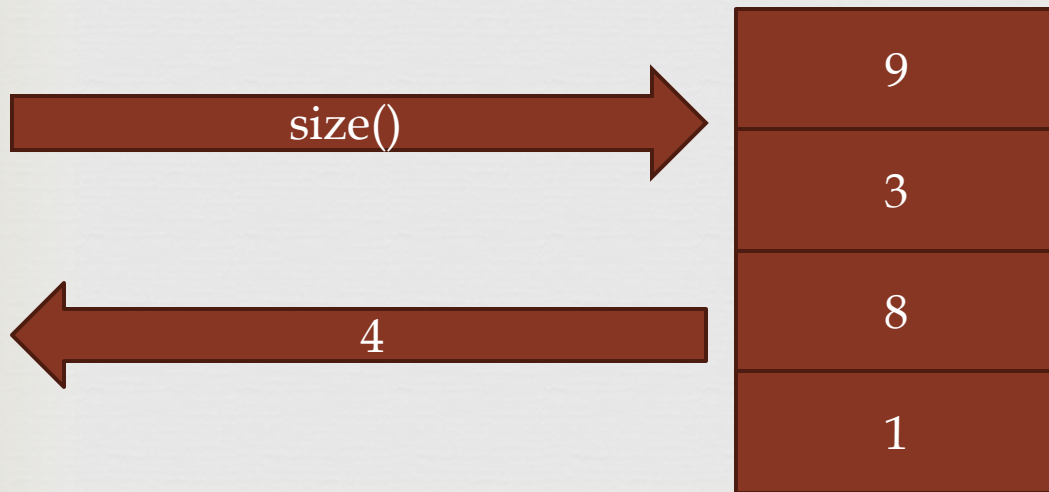


- ❧ Ao ser inserido um item na lista ele deve entrar no seu final

Listas



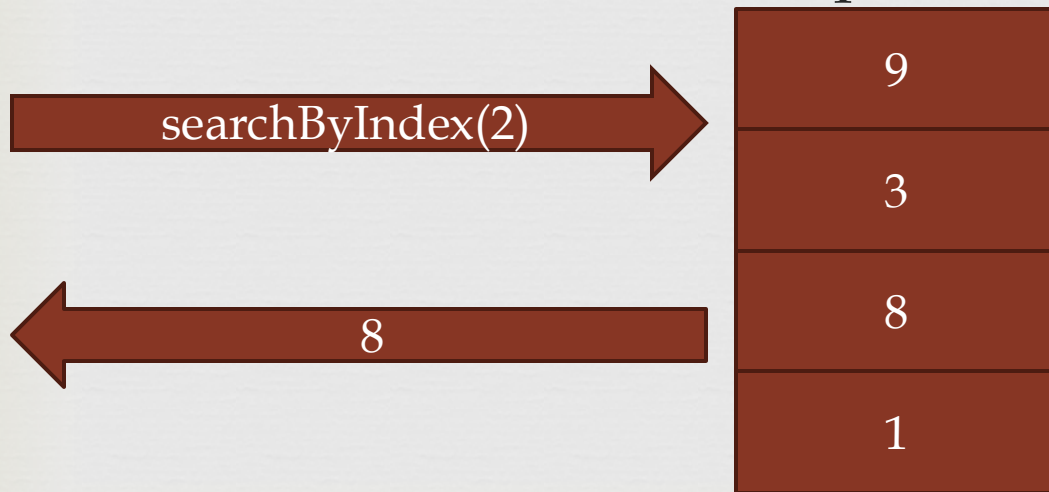
☞ Ao ser chamada `size` a função deve retornar o número de elementos da lista



Listas



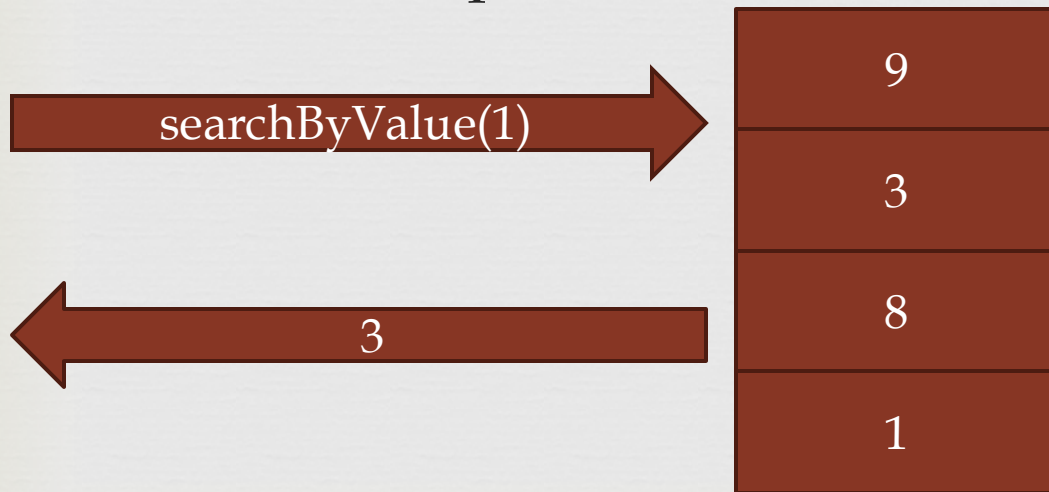
☞ Ao ser chamada `searchByIndex` a função deve retornar o elemento na posição desejada:



Listas



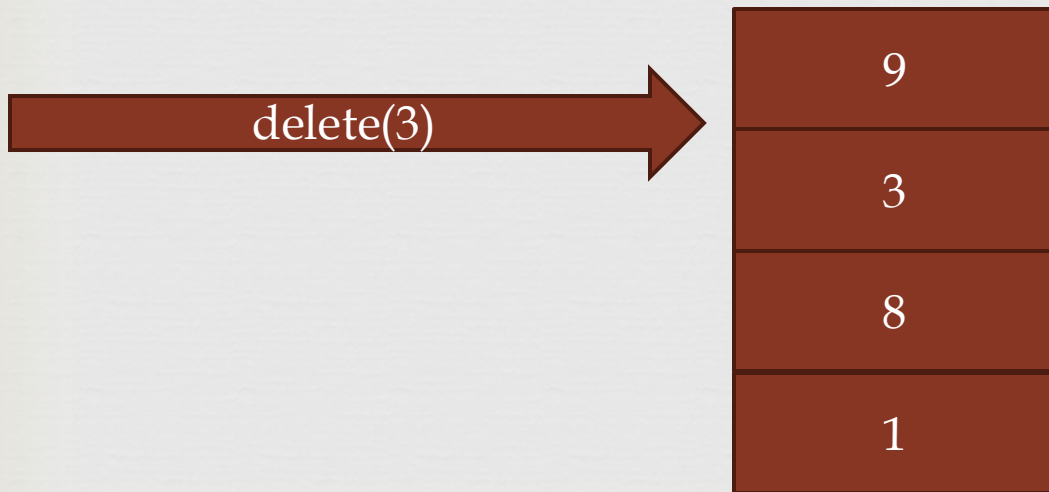
☞ Ao ser chamada `searchByValue` a função deve retornar a posição do elemento:



Listas



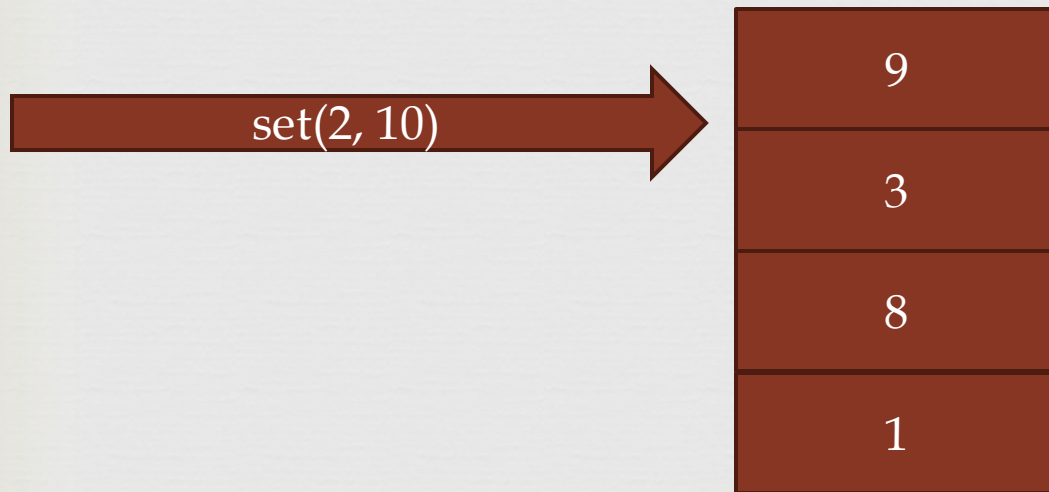
☞ Ao ser chamado delete, a função deve localizar o elemento e realocar os itens abaixo dele.



Listas



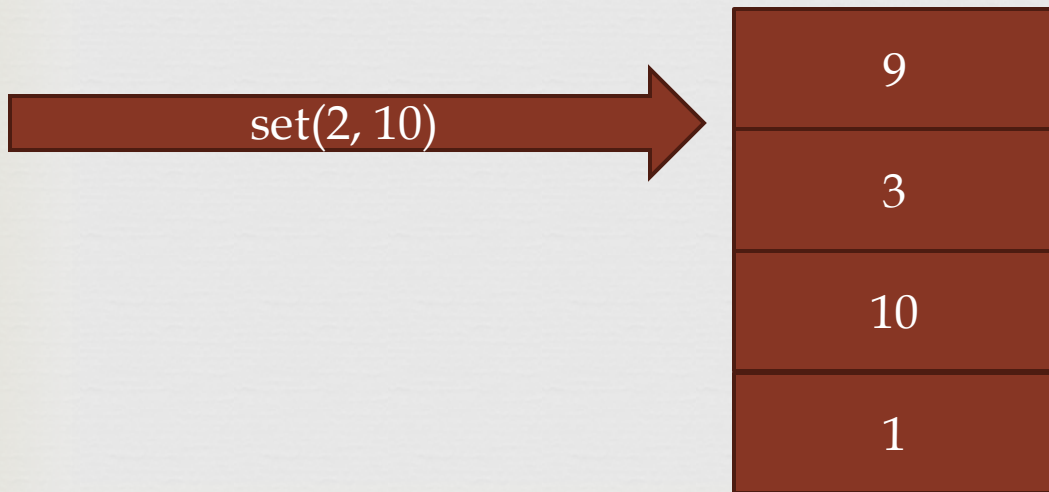
⌘ Ao ser chamado set, a função deve localizar o elemento e alterar seu valor se o índice existir.



Listas



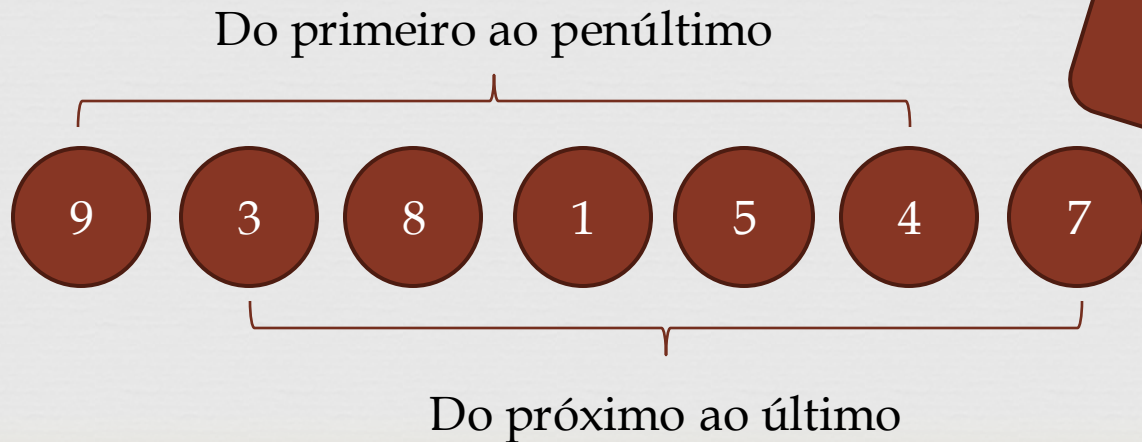
⌘ Ao ser chamado set, a função deve localizar o elemento e alterar seu valor se o índice existir.



Listas



☞ Ao ser chamado ordenar() a função deve ordenar os elementos da lista em ordem crescente



Trocando
quando
necessário

Listas

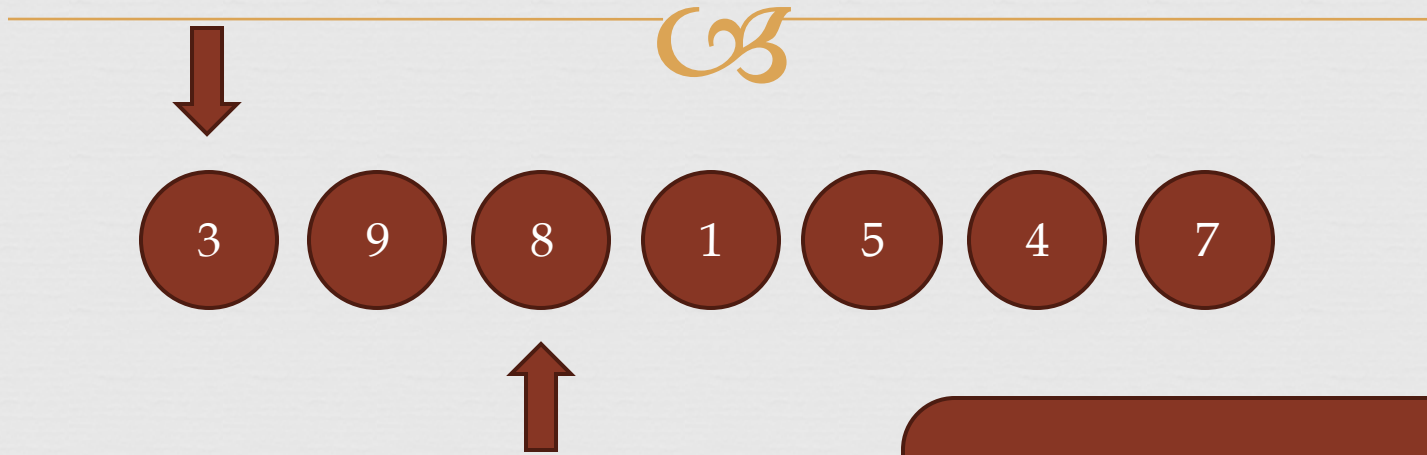


Listas



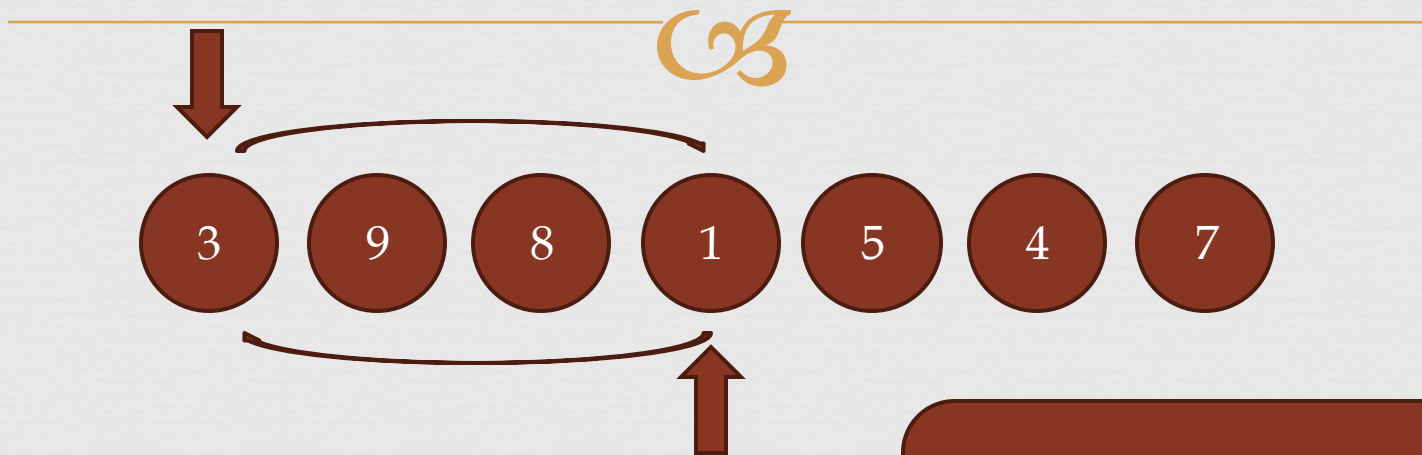
9 é maior que 3
Troca-se de posição

Listas



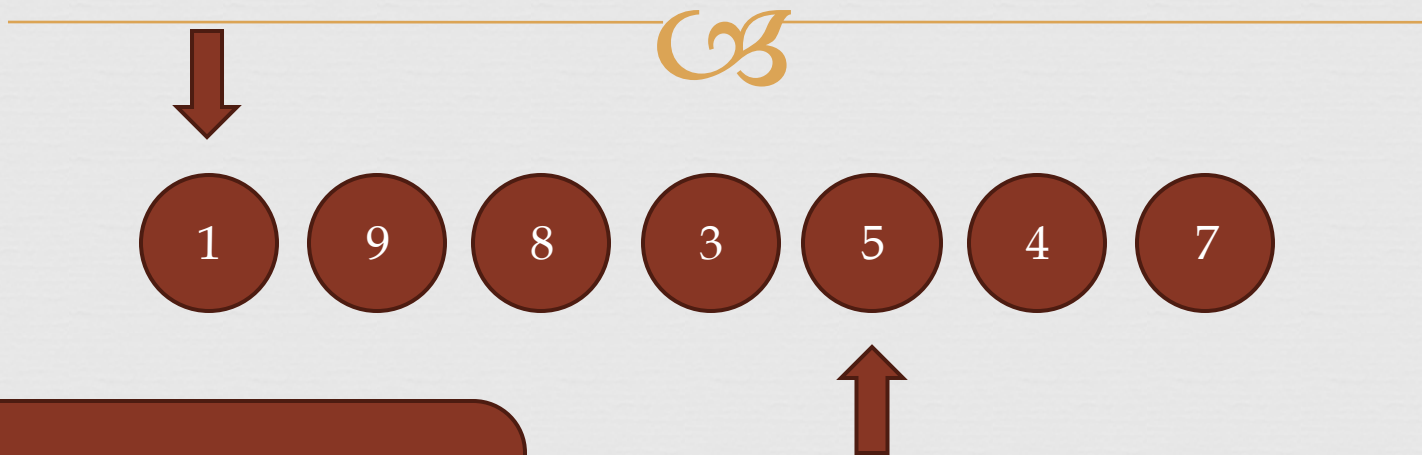
3 não é maior que 8
Deixa-se como está

Listas



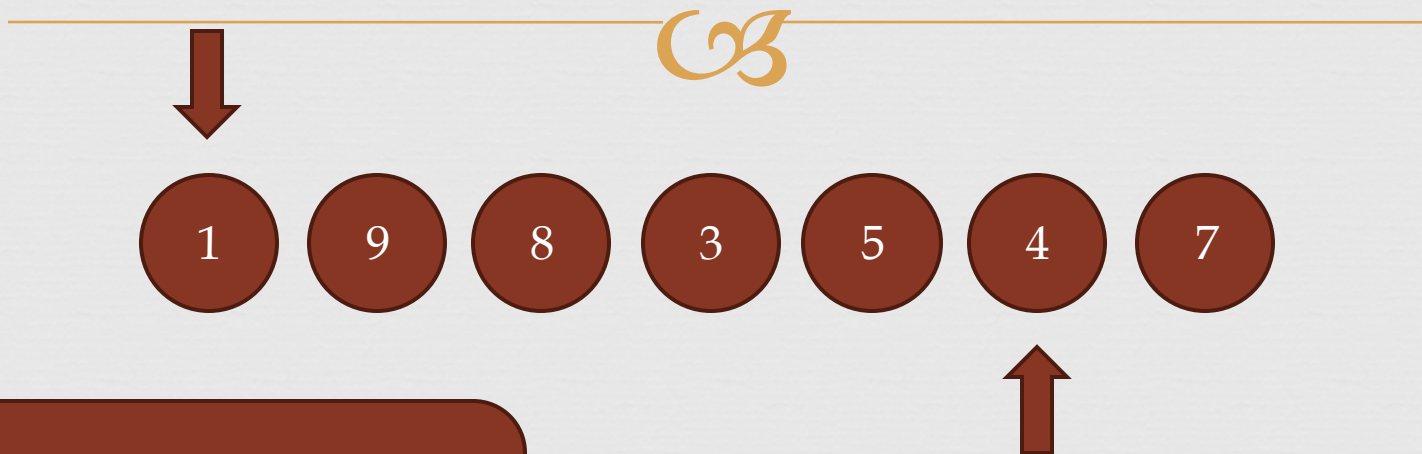
3 é maior que 1
Troca-se de posição

Listas



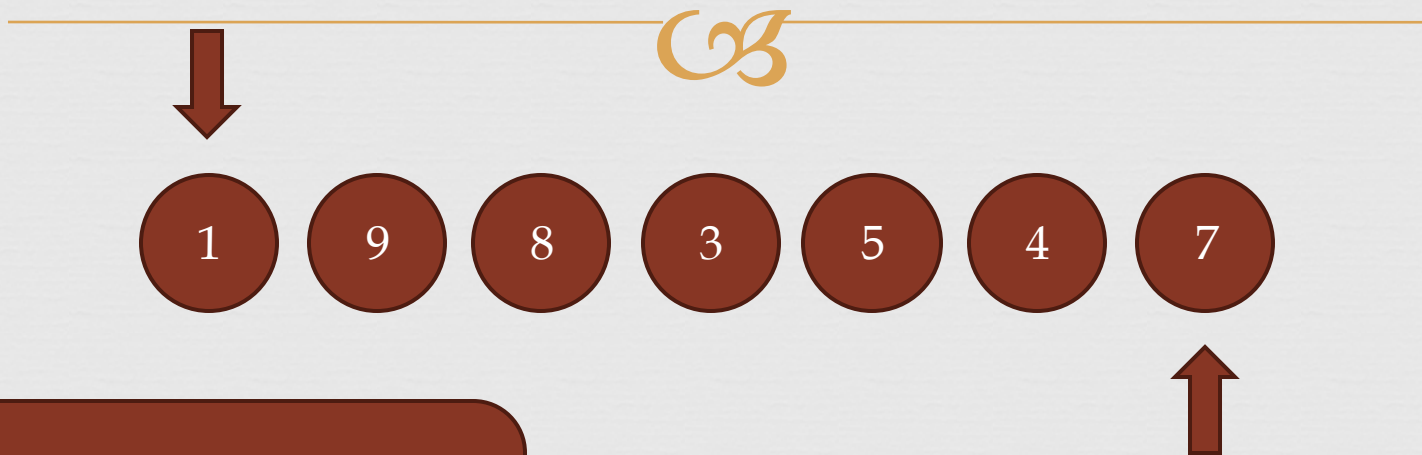
1 não é maior que 5
Deixa-se como está

Listas



1 não é maior que 4
Deixa-se como está

Listas



1 não é maior que 7
Deixa-se como está

Listas



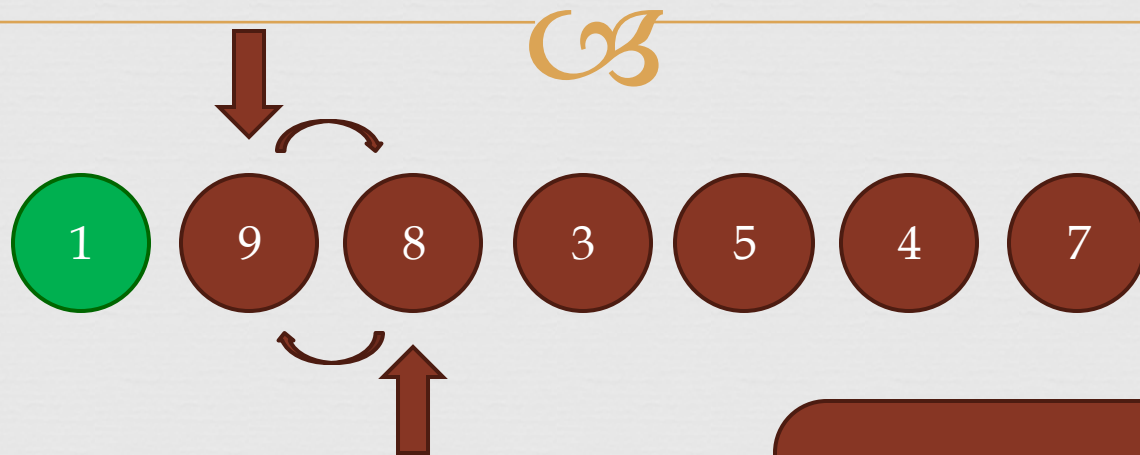
1 já está ordenado!

Listas



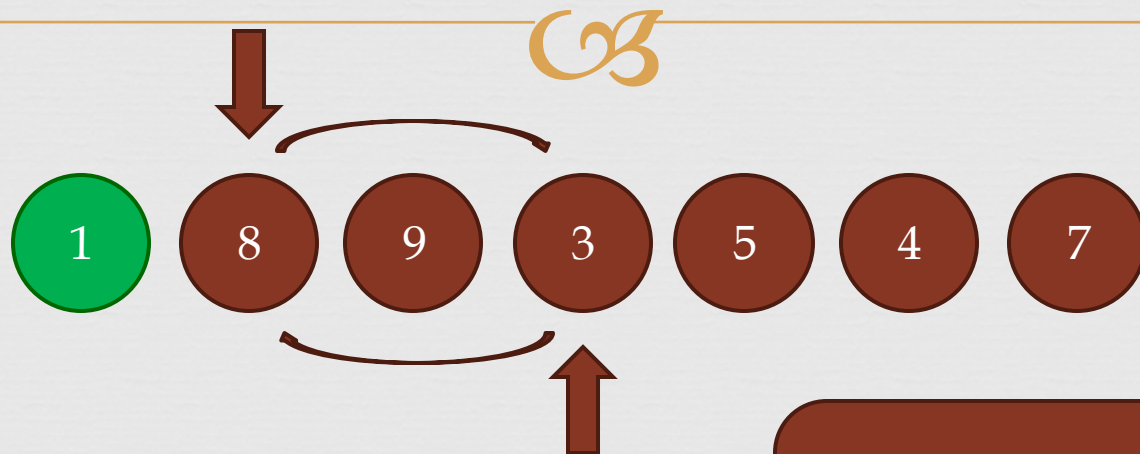
1 já está ordenado!

Listas



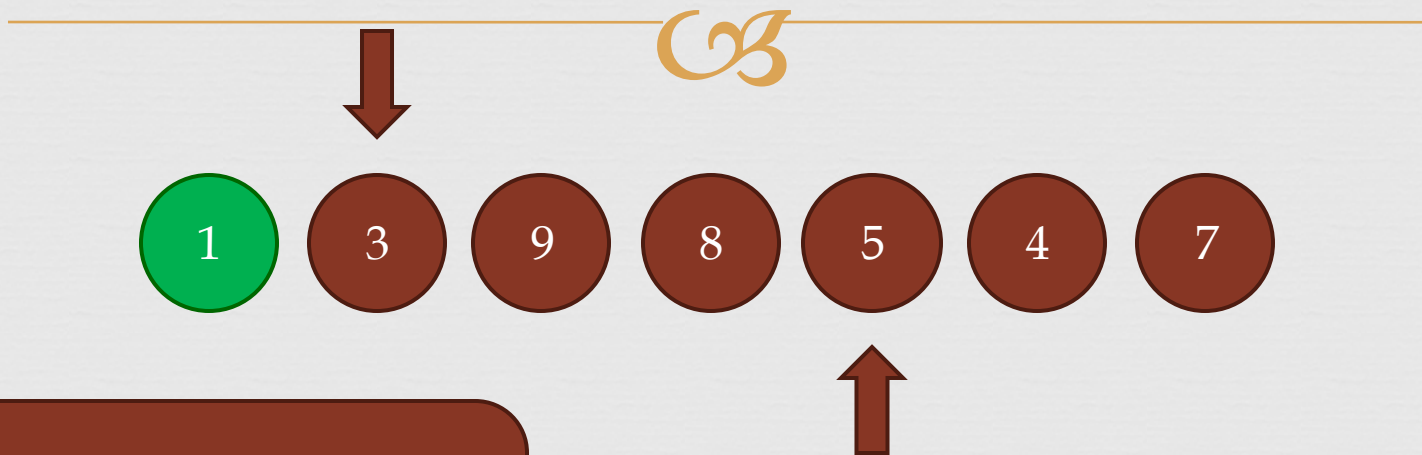
9 é maior que 8
Troca-se de posição

Listas



8 é maior que 3
Troca-se de posição

Listas



3 não é maior que 5
Deixa-se como está

Listas



3 não é maior que 4
Deixa-se como está

Listas



3 não é maior que 7
Deixa-se como está

Listas



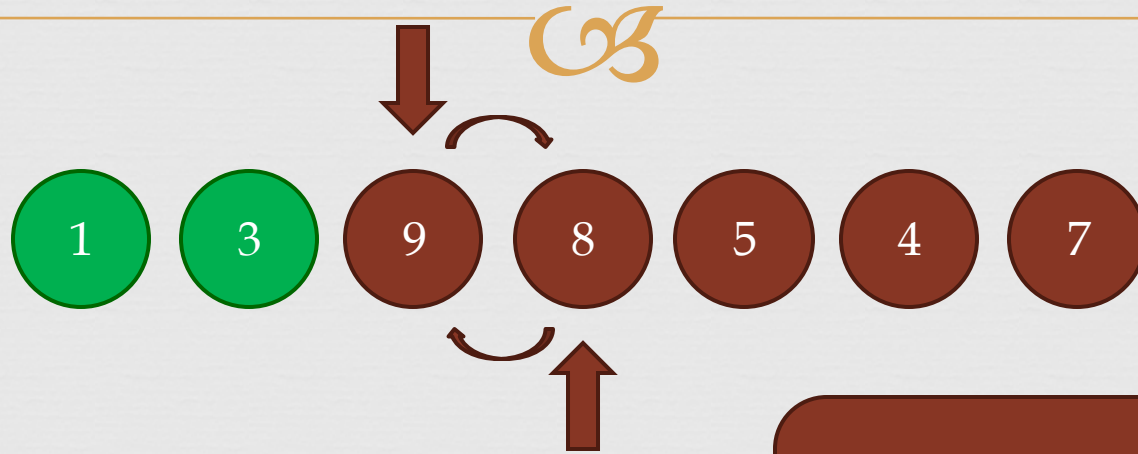
3 já está ordenado!

Listas



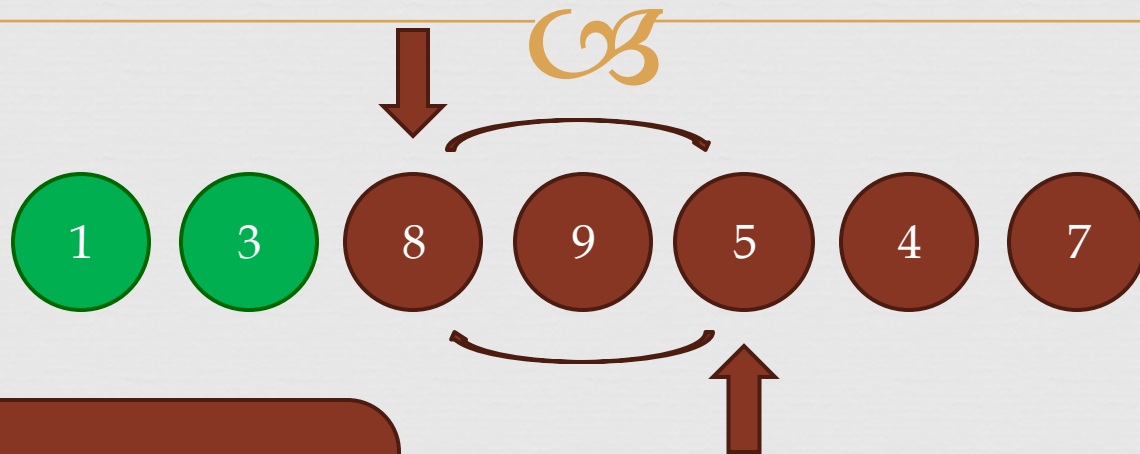
3 já está ordenado!

Listas



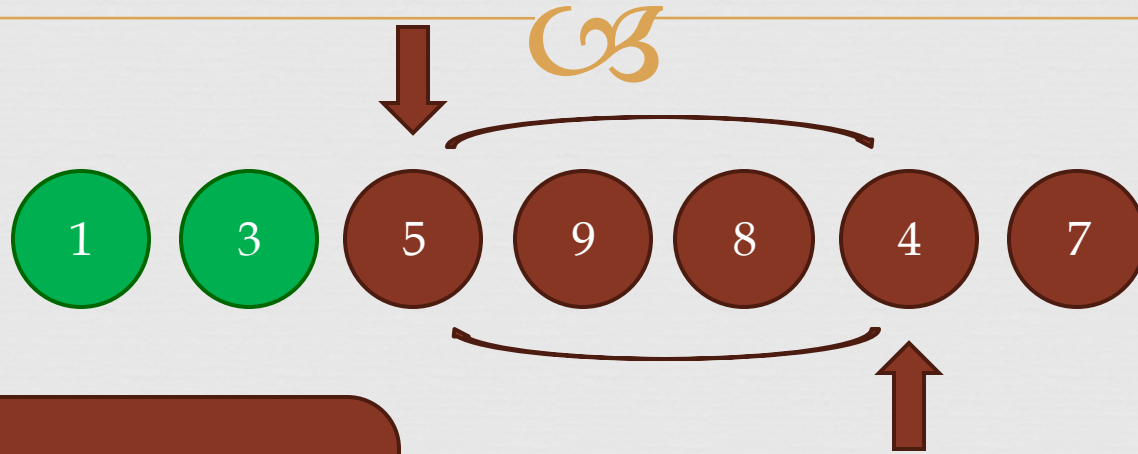
9 é maior que 8
Troca-se de posição

Listas



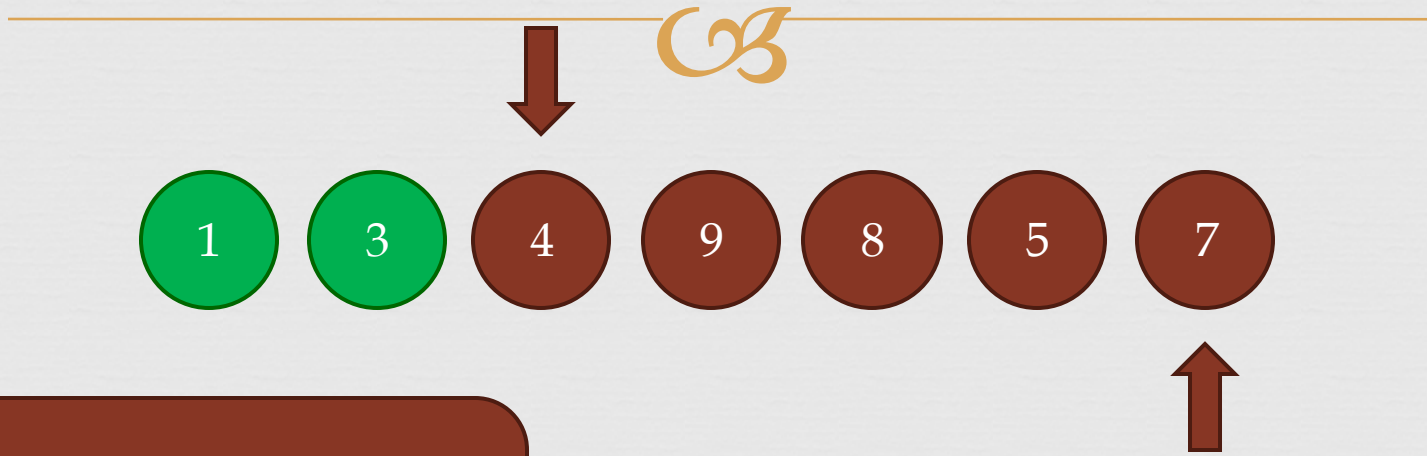
8 é maior que 5
Troca-se de posição

Listas



5 é maior que 4
Troca-se de posição

Listas



4 não é maior que 7
Deixa-se como está

Listas



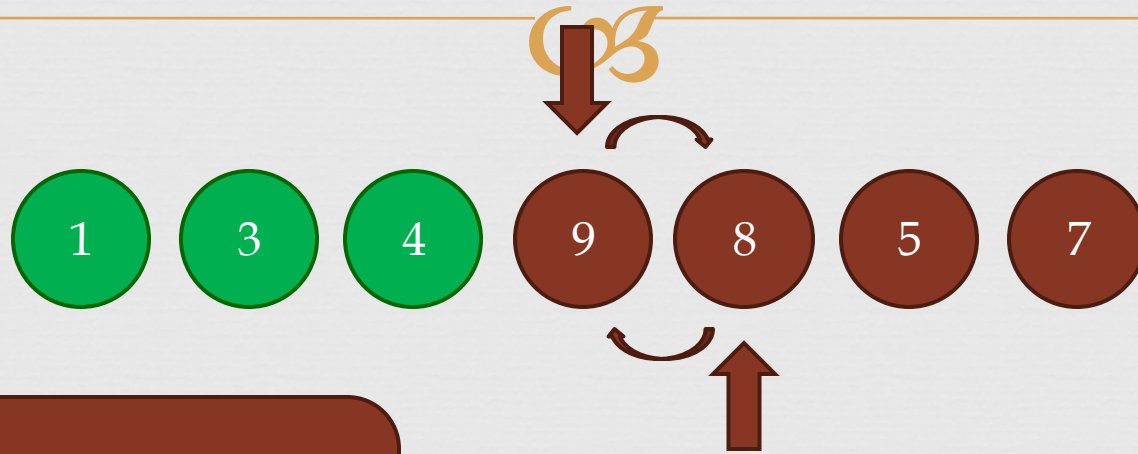
4 já está ordenado!

Listas



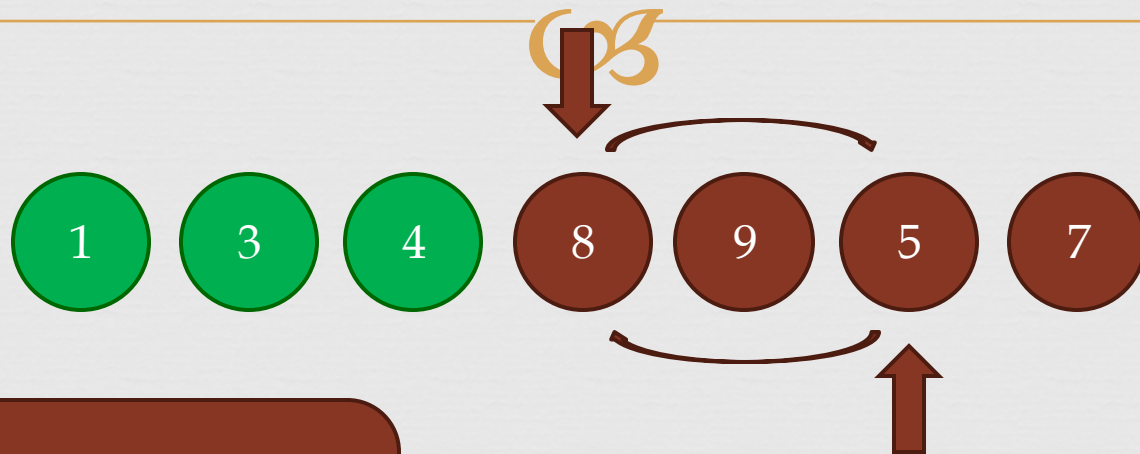
4 já está ordenado!

Listas



9 é maior que 8
Troca-se de posição

Listas



8 é maior que 5
Troca-se de posição

Listas



5 não é maior que 7
Deixa-se como está

Listas



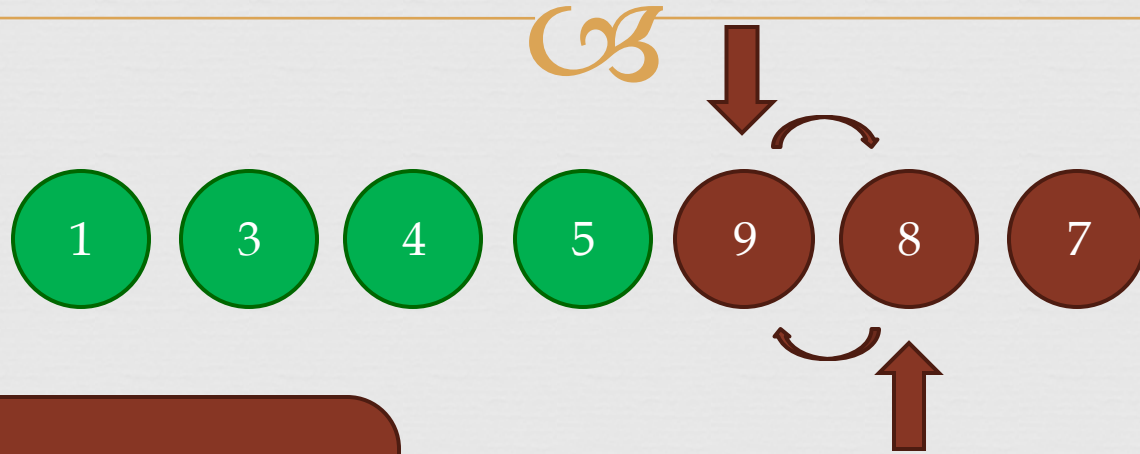
5 já está ordenado!

Listas



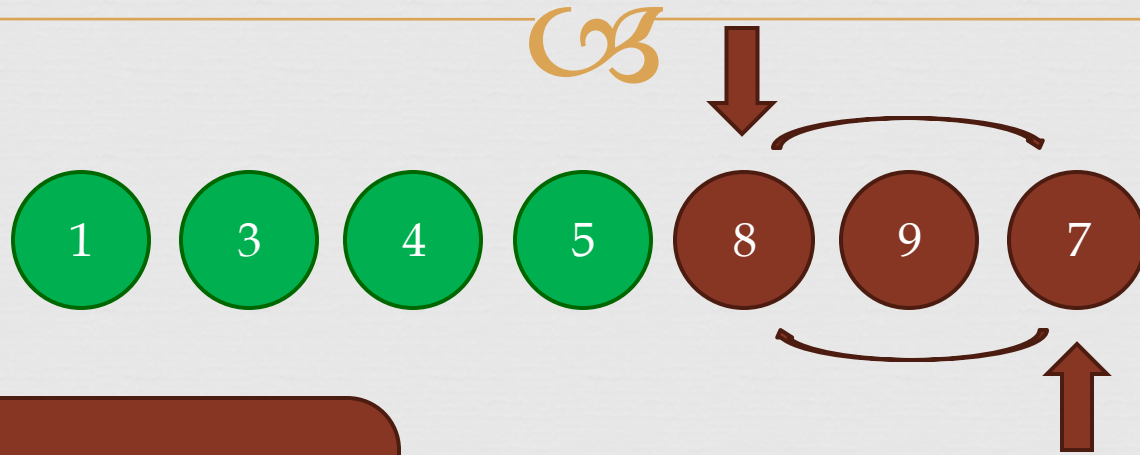
5 já está ordenado!

Listas



9 é maior que 8
Troca-se de posição

Listas



8 é maior que 7
Troca-se de posição

Listas



7 já está ordenado!

Listas



7 já está ordenado!

Listas



9 é maior que 8
Troca-se de posição

Listas



8 e 9 já estão ordenados!

Listas



8 e 9 já estão ordenados!

Listas

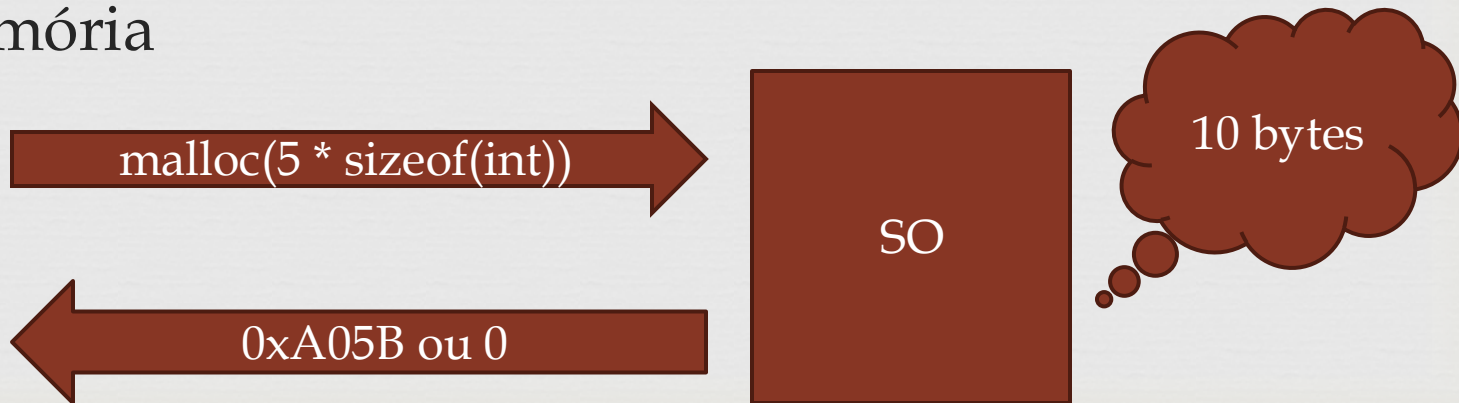


- ❧ Para fazer uma expansão de um vetor devemos trabalhar com alocação de memória
- ❧ Isto se faz usando as funções malloc, sizeof e free
- ❧ Mas como ela funciona?

Listas



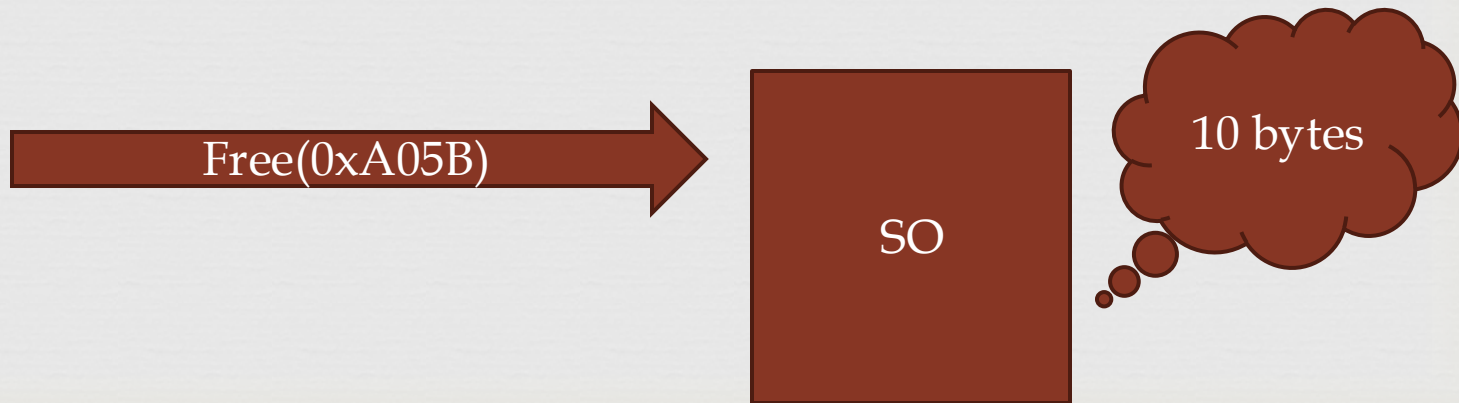
- Seu programa chama malloc, que pede ao SO por um número de bytes, vamos imaginar 5 inteiros
- O SO pode devolver um ponteiro ou 0 se não tiver memória



Listas



❧ Para liberar a memória devemos chamar free, passando como parâmetro o ponteiro para a memória



Listas



❧ O código abaixo mostra a alocação inicial do vetor

```
lista = malloc(tamanho * sizeof(int));  
if(!lista)  
{  
    printf("Erro de alocao!");  
    exit(-1);  
}
```

Listas



- ❧ Mas como se dá a expansão?
- ❧ Primeiro você recalcula o tamanho do novo vetor
- ❧ Aloca a memória necessária
- ❧ Copia os elementos para o novo vetor
- ❧ Libera a memória do vetor antigo
- ❧ Reposiciona o ponteiro da lista

Listas



lista

0	9
1	4
2	3
3	8
4	1

aux

0	
1	
2	
3	
4	
5	
6	
7	

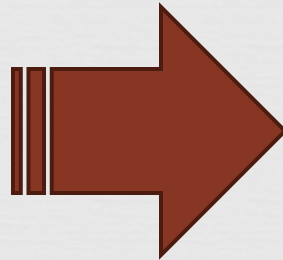
Calcula o novo
tamanho e
aloca a
memória

Listas



lista

0	9
1	4
2	3
3	8
4	1



0	
1	
2	
3	
4	
5	
6	
7	

aux

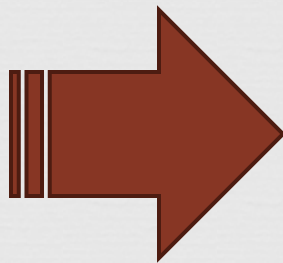
Copia os dados

Listas



lista

0	9
1	4
2	3
3	8
4	1



0	9
1	4
2	3
3	8
4	1
5	
6	
7	

aux

Copia os dados

Listas



lista



0	9
1	4
2	3
3	8
4	1
5	
6	
7	

aux



Libera a
memória do
vetor original

Listas



lista

aux

0	9
1	4
2	3
3	8
4	1
5	
6	
7	

Aponta o
ponteiro para a
nova posição