

Memória e passagem de parâmetros - Prática

Marco A L Barbosa

malbarbo.pro.br

1. Projete uma função que some um valor n a cada elemento de uma lista de números.
2. Projete uma função que receba como parâmetros uma lista e um índice i e modifique a lista removendo o elemento do índice i .

```
>>> lst = [7, 1, 8, 9]
>>> remove_indice(lst, 2);
>>> lst
[7, 1, 9]
# Escreva mais exemplos!
```

Dica: mova o elemento do índice i até o final e depois use `list.pop` para removê-lo. A função `list.pop` funciona da seguinte forma

```
>>> lst = [3, 9, 1, 2]
>>> lst.pop()
2
>>> lst
[3, 9, 1]
```

3. Projete uma função que receba como parâmetros uma lista, um índice i e um valor v , e modifique a lista inserindo o valor v no índice i . Dica: veja o exemplo `insere_ordenado`.
4. Projete uma função que remova todas as strings vazias de uma lista de strings.
5. Projete uma função que modifique uma lista colocando os elementos negativos antes dos positivos.
6. Ordenação por seleção é um algoritmo para ordenar uma lista de valores. A ideia do algoritmo é selecionar um valor mínimo da lista a partir da posição 0 e colocá-lo na posição 0, depois encontrar um valor mínimo da lista a partir da posição 1 e colocá-lo na posição 1, depois encontrar um valor mínimo da lista a partir da posição 2 ... e assim por diante. Por exemplo, vamos considerar a lista `[8, 5, 4, 1, 2]`.

O valor mínimo a partir da posição 0 é 1 (que está no índice 3), colocando 1 na posição 0, obtemos `[1, 5, 4, 8, 2]`.

O valor mínimo a partir da posição 1 é 2 (que está no índice 4), colocando 2 na posição 1, obtemos `[1, 2, 4, 8, 5]`.

O valor mínimo a partir da posição 2 é 4 (que está no índice 2), colocando 4 na posição 2, obtemos `[1, 2, 4, 8, 5]`.

O valor mínimo a partir da posição 3 é 5 (que está no índice 4), colocando 5 na posição 3, obtemos `[1, 2, 4, 5, 8]`.

Baseado nesta descrição, projete uma função que faça a ordenação dos valores usando o algoritmo de ordenação por seleção.