

Exercícios - Pilha e Fila

Estrutura de dados 2024.1

Pilha

Considere que você tem acesso a uma TAD **Pilha** com as seguintes funcionalidades implementadas:

```
1  criar_pilha(); // Retorna uma pilha inicialmente vazia
2  acessar(P) // Retorna o elemento no topo da pilha P
3  inserir(P, chave) // Insere a chave no topo da pilha P
4  remover(P) // Remove o elemento do topo da pilha P
5  vazia(P) // Verifica se há algum elemento na pilha P
```

Problema 1 Considere o seguinte código que utiliza a pilha:

```
1  p = Pilha();
2  inserir(p, 4)
3  inserir(p, -5)
4  remover(p)
5  inserir(p, 6)
6  inserir(p, 10)
7  inserir(p, 2)
8  remover(p)
9  remover(p)
10 inserir(p, 6)
11 remover(p)
12 remover(p)
13 remover(p)
```

Liste os elementos da pilha na ordem em que foram removidos.

Problema 2 Escreva o código de uma função **esvaziar(p)** que recebe como parâmetro uma pilha. A função deve remover todos os elementos da pilha p, tornando-a vazia.

Problema 3 Escreva o código de uma função **tamanho(p)** que recebe como parâmetro uma pilha. A função deve retornar a quantidade de elementos na pilha p. No final da função, a pilha deve estar no mesmo estado inicial, ou

seja, com os mesmos elementos de quando a função foi chamada. **Você deve utilizar apenas outras pilhas como estruturas auxiliares.**

Problema 4 Escreva o código de uma função **inverte_pilha(p1)** que recebe uma pilha p1. Seu objetivo é inverter o conteúdo da pilha p1, **usando apenas pilhas auxiliares.** Você não deve usar nenhuma outra estrutura, como listas.

Problema 5 Escreva o código de uma função **palindromo(lista_enc)** que recebe uma **lista encadeada simples**. Seu objetivo é checar se a lista encadeada é palíndromo ou não. Complexidade esperada: $O(n)$.

Problema 6 Escreva o código de uma função **verifica_parenteses(S)** que recebe como entrada uma string S, composta apenas por parênteses, ou seja, '(' e ')'. A função deve retornar *true* se os parênteses foram abertos e fechados na ordem correta, ou *false* caso contrário.

Por exemplo, as strings "()", "(()())", "(()())" estão corretas. Enquanto "(", "()", "(()())" e "(()())(" não estão.

Fila

Problema 7 Considere a seguinte definição de fila sequencial.

```
1 class Fila:
2     def __init__(self):
3         self.capacidade = 8
4         self.inicio = 0
5         self.tamanho = 0
6         self.elementos = [None] * self.capacidade
```

Implemente as seguintes funcionalidades de uma **fila sequencial**:

- int acessar(fila)
- void inserir(fila, chave)
- void remover(fila)

Problema 8 Considere a seguinte definição de fila encadeada.

```
1 class No:
2     def __init__(self, chave: int, proximo):
3         self.chave = chave
4         self.proximo = proximo
5
6 class Fila:
7     def __init__(self):
8         self.cabeca = None
9         self.cauda = None
```

Problema 9 Implemente as seguintes funcionalidades de uma **fila sequencial**:

- int acessar(fila)
- void inserir(fila, chave)
- void remover(fila)

Problema 10 Considere o seguinte código que utiliza a fila:

```
1 Fila* f = criar_fila();
2 inserir(f, 2)
3 inserir(f, -7)
4 remover(f)
5 inserir(f, 5)
6 inserir(f, 8)
7 inserir(f, 20)
8 remover(f)
9 remover(f)
10 inserir(f, 1)
11 remover(f)
12 remover(f)
13 remover(f)
```

Liste os elementos da fila na ordem em que foram removidos.