Atividade – Pilhas (PUSH, SIZE, TOP, POP, EMPTY)

Objetivo: Praticar o uso de estruturas de pilha por meio de problemas reais e desafios de lógica.

Exercício 1 – PUSH

Um navegador salva o histórico das páginas visitadas usando uma pilha. Sempre que você acessa uma nova página, ela é empilhada.

Tarefa: Simule o acesso a 3 páginas diferentes e mostre a pilha de histórico.

Código de exemplo:

```
historico = []
historico.append("google.com")
historico.append("youtube.com")
historico.append("github.com")
print("Histórico de navegação (PUSH):", historico)
```

Exercício 2 – SIZE

Uma impressora trabalha com uma pilha de documentos de emergência. Cada novo documento é empilhado. Antes de iniciar, o sistema precisa saber quantos documentos tem na pilha.

Tarefa: Mostre o número de documentos aguardando impressão.

Código de exemplo:

```
documentos = ["Contrato.pdf", "Relatório.docx", "Proposta.xlsx"]
print("Número de documentos na fila (SIZE):", len(documentos))
```

Exercício 3 – TOP

Um editor de texto permite desfazer ações. Cada ação feita é empilhada. Quando o usuário clica em 'Desfazer', ele visualiza a última ação.

Tarefa: Mostre qual foi a última ação feita (sem remover ainda).

Código de exemplo:

```
acoes = ["digitar 'Olá'", "negrito", "inserir imagem", "centralizar
texto"]
print("Última ação feita (TOP):", acoes[-1])
```

Exercício 4 – POP

Durante o desenvolvimento, um programador quer desfazer as 3 últimas alterações que ele empilhou como ações. Para isso, ele vai usando POP para desfazer.

Tarefa: Remova as últimas 3 ações usando POP.

Código de exemplo:

```
alteracoes = ["adicionar função", "renomear variável", "ajustar
identação", "remover comentário"]
print("Desfazendo:", alteracoes.pop())
print("Desfazendo:", alteracoes.pop())
print("Desfazendo:", alteracoes.pop())
print("Alterações restantes:", alteracoes)
```

Exercício 5 – EMPTY

Um sistema técnico lida com uma pilha de chamados urgentes. Se a pilha estiver vazia, ninguém precisa de atendimento agora.

Tarefa: Verifique se há chamados pendentes ou se está vazio.

Código de exemplo:

```
chamados = []
if len(chamados) == 0:
    print("Nenhum chamado urgente no momento. (EMPTY)")
else:
    print("Chamados aguardando resolução!")
```

Demais Exercícios

Exercício 1 – PUSH com input()

O usuário digita 3 números para empilhar. Após isso, o programa deve mostrar a pilha resultante.

Código de exemplo:

```
pilha = []
for i in range(3):
    numero = int(input("Digite um número para empilhar (PUSH): "))
    pilha.append(numero)
print("Pilha após os PUSHs:", pilha)
```

Exercício 2 – SIZE após operações

Mostre o tamanho final da pilha após empilhar dois valores e remover um.

Código de exemplo:

```
pilha = [7, 14, 21]
pilha.append(28)
pilha.append(35)
pilha.pop()
print("Tamanho final da pilha (SIZE):", len(pilha))
```

Exercício 3 – Ver TOP sem alterar a pilha

Mostre o topo da pilha e prove que ela não foi alterada.

Código de exemplo:

```
pilha = [1, 3, 5, 7, 9]
print("Pilha:", pilha)
print("Topo da pilha (TOP):", pilha[-1])
print("Pilha depois do TOP (sem alterações):", pilha)
```

```
Obs.: ver o topo, pilha[-1], não altera a pilha, pilha.pop() remove e
retorna.
```

Elemento	Índice Positivo	Índice Negativo
1	0	-5
3	1	-4
5	2	-3
7	3	-2
9	4	**-1**

Exercício 4 – POP até esvaziar a pilha

Desempilhe todos os elementos de uma pilha e mostre a pilha a cada remoção.

Código de exemplo:

```
pilha = [100, 200, 300]
while len(pilha) > 0:
    elemento = pilha.pop()
    print("Elemento desempilhado:", elemento)
    print("Pilha agora:", pilha)
```

Obs.: enquanto a pilha tiver elementos (ou seja, enquanto o tamanho dela for maior que zero), continue repetindo o bloco de código.

Exercício 5 – Verificar EMPTY após remoções

Desempilhe todos os elementos e verifique se a pilha está vazia.

Código de exemplo:

```
pilha = [11, 22]
pilha.pop()
pilha.pop()
if len(pilha) == 0:
    print("A pilha está vazia (EMPTY).")
else:
    print("Ainda tem elementos na pilha!")
```