

Lista de exercícios

– Filas e Pilhas –

1. Considere que as seguintes operações foram efetuadas em uma pilha: **push(d)**, **push(h)**, **pop()**, **push(f)**, **push(s)**, **pop()**, **pop()**, **push(m)**.
 - (a) Assuma que a pilha está inicialmente vazia. Qual é a sequência de valores que são removidos utilizando a função **pop()**? Qual é o estado final da pilha? Ordene pelos valores do topo ao final.
 - (b) Assuma que você tenha de trocar o comportamento desta pilha para uma fila. Qual seria o estado final da fila? Ordene pelos valores da frente ao final.
2. Valide as expressões abaixo utilizando o conceito de pilhas. Considere que para cada chave, colchete ou parênteses aberto, a pilha faz um **push()** e, para cada fechado, a pilha faz um **pop()**. Indique se a expressão é válida observando estes sinais de associação. Para cada operação, indique o estado atual da pilha ordenando os valores do topo ao final.
 - (a) $[a + \{ b / (c - d) + e / (f + g) \} - h]$
 - (b) $[a \{ b + [c (d + e) - f] + g \}$
3. Você deve inserir os números de 1 a 5 – obrigatoriamente nessa ordem – em uma pilha. É permitido realizar **pop()** a qualquer momento. Através destas operações, indique a ordem das operações de **push()** e **pop()** que devem ocorrer para que a ordem dos números removidos da pilha (popped) seja:
 - (a) 2, 4, 5, 3, 1
 - (b) 1, 2, 5, 4, 3
 - (c) 1, 3, 5, 4, 2

4. No exercício a seguir, considere que você possui três pilhas, sendo elas **s1**, **s2** e **s3**, com o estado inicial aparecendo na coluna **Começo** e o estado final aparecendo na coluna **Fim**. Encontre a sequência de **push()** e **pop()** que leve as pilhas para o estado final.

Instruções: Para fazer, por exemplo, o **pop()** do primeiro elemento de **s1** e colocá-lo em **s3**, faça: **s3.push(s1.pop())**.

Começo			Fim		
A					A
B					B
C					D
D					C
—	—	—	—	—	—
s1	s2	s3	s1	s2	s3

5. Adapte o programa do exercício 4 para que as operações levem ao seguinte estado final:

Começo			Fim		
A					B
B					D
C					A
D					C
—	—	—	—	—	—
s1	s2	s3	s1	s2	s3

6. Considere que as seguintes operações foram efetuadas em uma pilha: **push(a)**, **push(b)**, **push(c)**, **pop()**, **push(d)**, **push(e)**, **pop()**, **pop()**, **pop()**.
- Assuma que a pilha está inicialmente vazia. Qual é a sequência de valores que são removidos utilizando a função **pop()**? Qual é o estado final da pilha? Ordene pelos valores do topo ao final.
 - Qual será a ordem dos elementos removidos se considerarmos uma **fila**?
 - Considerando a mesma pilha inicial e sem modificar a ordem das operações **push()**, altere a ordem das operações **pop()** para que, ao final, os elementos tenham sido removidos na ordem: **b, d, c, a, e**.
7. Considere a inserção das letras RESPONSABILIDADE (nesta ordem) em uma pilha que armazena dados do tipo *char*. Entre as inserções, são feitas remoções e, a cada remoção, a letra removida é armazenada em uma sequência. Escreva a sequência de inserções e remoções para que as letras sejam retiradas da pilha na seguinte ordem: SOPENBIALSADDIER.

Respostas:**Exercício 1:**

a) Sequência dos elementos removidos (pop): h,s,f

Estado final da pilha: m, d

b) Sequência dos elementos removidos (desenfileirar): d,h,f

Estado final da lista: s, m

Exercício 2:**a)**

- (pilha vazia)

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

Pilha vazia ao final, portanto, expressão válida.

b)

- (pilha vazia)

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

{

Pilha não vazia ao final, portanto, expressão inválida.

Exercício 3:

	Estado Final		
	24531	12543	13542
push(1)	push(1)	push(1)	push(1)
push(2)	pop	pop	pop
pop	push(2)	push(2)	push(2)
push(3)	pop	pop	push(3)
push(4)	push(3)	pop	pop
pop	push(4)	push(4)	push(4)
push(5)	push(5)	push(5)	push(5)
pop	pop	pop	pop
pop	pop	pop	pop
pop	pop	pop	pop

Exercício 4:

s2.push(s1.pop())

s2.push(s1.pop())

s3.push(s1.pop())

s3.push(s1.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

s3.push(s2.pop())

Exercício 6:**a)** c, e, d, b, a**b)** a, b, c, d, e**c)** push(a), push(b), pop(), push(c), push(d), pop(), pop(), pop() push(e), pop()**Exercício 7** push(R), push(E), push(S), pop() push(P), push(O), pop(), pop(), pop(), push(N), pop()

push(S), push(A), push(B), pop(), push(I), pop(), pop(), push(L), pop(), pop(), push(I), push(D), push(A),

pop(), push(D),pop(), pop(), pop(), push(E), pop(), pop()