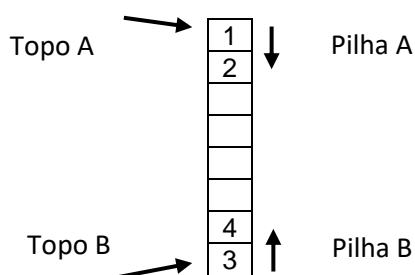




Lista de Exercícios 1

QUESTÃO 1: (ENADE 2014, Adaptada) Uma estrutura de dados do tipo pilha pode ser usada em um algoritmo que permite imprimir uma palavra de forma invertida. Por exemplo, ABACAXI deve ser impresso IXACABA. Utilizando uma pilha e suas operações base, desenvolva a função inverte que recebe uma String e, utilizando as funções push e pop da pilha, retorne a palavra de forma invertida.

QUESTÃO 2. Uma Pilha pode ser definida como um conjunto ordenado de dados, no qual os dados podem ser inseridos ou removidos a partir de uma extremidade chamada topo da pilha. Duas pilhas A e B podem ser implementadas a partir do compartilhamento do mesmo vetor de tamanho determinado, como apresentado na figura a seguir. Escreva as seguintes funções em *Javascript*.



- Declarações de constantes e procedimento de inicializar as pilhas (os valores de topoA e topoB).
- As funções *ÉVaziaA* e *ÉVaziaB*.
- As funções *empilhaA*, *empilhaB*, *desempilhaA* e *desempilhaB*. Só deve ser emitida uma mensagem de pilha cheia se todas as posições do vetor estiverem ocupadas.

QUESTÃO 3. Implemente uma função que troca de lugar o elemento que está no topo da pilha com o elemento que está na base da pilha. Usar apenas uma pilha como auxiliar.

QUESTÃO 4. Implemente um conversor de números decimais para binários utilizando uma pilha.

QUESTÃO 5. É possível analisar se uma sequência parêntesis e colchetes está bem-formada, ou seja, parênteses e colchetes são fechados na ordem inversa àquela em que foram abertos utilizando uma pilha. Por exemplo, a sequência `[() [()]] ()` está bem-formada, enquanto `(()]` está malformada. Implemente um programa que recebe um texto e analisa se ele está bem formado em termos de ordem de parêntesis e colchetes.

QUESTÃO 6. A Notação Polonesa Reversa (RPN, do inglês Reverse Polish Notation) é uma notação matemática onde cada operador segue todos os seus operandos. Por exemplo, para somar três e quatro, alguém escreveria "3 4 +" em vez de "3 + 4". Se houver múltiplas operações, o operador é dado imediatamente após seu segundo operando; então a expressão "3 - 4 + 5" seria escrita como "3 4 - 5 +", primeiro subtraindo 4 de 3 e depois adicionando 5 a isso. Desenvolva um método que transforme a expressão algébrica com parênteses em uma forma RPN. Você pode supor que para os casos de teste abaixo apenas letras simples serão usadas e colchetes `[]` não serão utilizados. A ordem de precedência de operadores é muito importante e não deve ser ignorada.

Entrada	Saída
$(a+(b*c))$	abc^*+
$((a+b)*(z+x))$	$ab+zx+^*$
$((a+t)*((b+(a+c))^{(c+d)}))$	$at+bac++cd+^*$
$a+b*c-d$	abc^*+d-
$(a+b)+c/d$	$ab+cd/+$
$a*b-(c-d)+e$	ab^*cd-e+

QUESTÃO 7. Dada uma pilha contendo números inteiros, implemente uma função que remova todos os elementos duplicados, mantendo apenas a primeira ocorrência de cada número (mais próxima do topo). A solução deve preservar a ordem relativa dos elementos originais.

Entrada:

Uma lista de números representando a pilha (o último elemento da lista representa o topo).

Saída:

Uma lista representando a pilha resultante após a remoção de duplicatas.

Exemplo:

Entrada: [3, 7, 3, 2, 7, 1, 4, 2]

Saída: [3, 7, 2, 1, 4]

QUESTÃO 8. Imagine que você tem um conjunto de pratos empilhados. No entanto, por questões de organização, cada pilha só pode conter um número máximo **capacidade** de pratos. Se uma pilha atingir essa capacidade, uma nova pilha é criada.

Implemente uma estrutura de dados PilhaDePratos que suporte as operações:

`empilha(int x)`: Adiciona um prato à última pilha disponível ou cria uma nova pilha, se necessário.

`desempilha()`: Remove um prato da última pilha disponível e retorna seu valor. Se todas as pilhas estiverem vazias, retorne None.

Entrada e Saída Esperadas:

`PilhaDePratos(3)` // Define a capacidade de cada pilha como 3

`empilha(5)`

`empilha(10)`

`empilha(15)`

`empilha(20)` // Nova pilha criada

`desempilha()` // Retorna 20

`desempilha()` // // Retorna 15