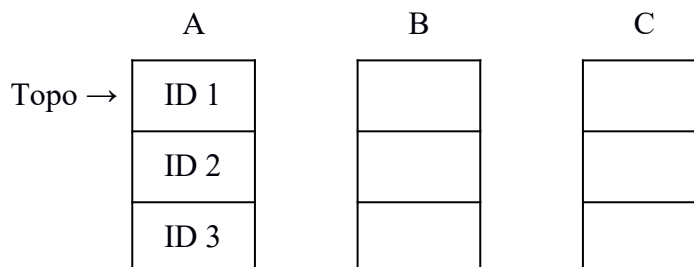


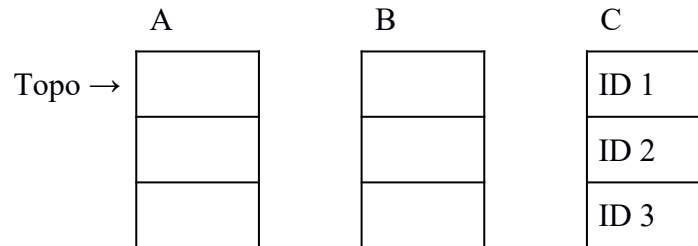
Lista de Exercícios Número 2 - Pilhas

1. O que é e para que serve uma pilha?
2. Desenvolva uma função para testar se uma pilha P1 tem mais elementos que uma pilha P2. Considere que as pilhas P1 e P2 são implementadas usando implementação encadeada dinâmica. Considere ainda o TAD Pilha especificado em sala de aula.
3. Implemente uma função para inverter a posição dos elementos de uma pilha sequencial dinâmica. Você pode utilizar o TAD pilha visto em sala.
4. Um determinado estacionamento de carros tem o formato de corredor e possui capacidade para dez carros. Infelizmente, há apenas uma passagem de acesso aos carros, localizada em uma das extremidades do corredor. Quando um cliente decide retirar um carro, os outros carros impedindo sua saída deverão ser retirados antes. Após o cliente retirar seu carro, os outros carros são estacionados novamente, mantendo a ordem em que estavam antes da remoção daquele carro. Escreva o TAD e implemente as operações necessárias para processar o fluxo de chegada/saída deste estacionamento (`int estaciona_carro(char* placa) / int retira_carro(char* placa)`), considerando que um carro é identificado pela placa. O valor de retorno das funções `estaciona_carro` e `retira_carro` indicam se a operação teve sucesso (valor 1) ou falhou (valor 0).
5. Considere três pilhas A, B e C, e os comandos `void empilha(Pilha *p, Elemento e)` e `Elemento *desempilha(Pilha *p)` que, como os nomes sugerem, servem para empilhar e desempilhar elementos de uma determinada pilha. Supondo que as pilhas estão inicialmente na seguinte configuração:



(continua na próxima página)

e que, em nenhum momento um elemento com ID maior pode estar empilhado sobre um elemento de ID menor. Descreva a sequência de comandos (utilize a menor quantidade de operações possível) empilha/desempilha necessários para que a configuração de A, B e C passe a ser:



6. Escreva um algoritmo para determinar se uma string de caracteres de entrada é da forma $a^n b^n$ tal que n é um número inteiro maior ou igual zero. Exemplo de strings na forma $a^n b^n$: ab, aabb, aaabbb. Não é permitido realizar qualquer tipo de contagem de caracteres da string de entrada.
7. Implemente uma calculadora polonesa usando pilha com alocação estática. O programa deve ler um arquivo com uma sequência de expressões em notação polonesa, calcular e imprimir os resultados (na saída padrão). Notação Polonesa é uma forma de notação para lógica, aritmética e álgebra em que os operadores devem preceder os valores, a notação polonesa de soma é: $+ 1 2$. Quanto mais "interna" a operação, antes ela deve ser executada. Então, as expressões: $* + 1 2 3$ seria calculado como $(1 + 2) * 3$; $* 3 + 1 2$ seria calculado como $3 * (1 + 2)$
8. Escreva um algoritmo que use uma pilha para inverter a ordem das letras de cada palavra de uma string, preservando a ordem das palavras. Por exemplo, dado o texto ESTE EXERCICIO E MUITO FACIL a saída deve ser ETSE OICICREXE E OTIUM LICAF.
9. Na seguinte sequência, uma letra significa '*empilha(letra)*' e um asterisco significa '*desempilha*'. Dê a sequência de valores devolvidos pelas operações *desempilha* quando esta sequência de operações é realizada em uma pilha inicialmente vazia.

E A S * Y * Q U E * * * S T * * * I O * N * * *

10. Escreva o TAD de uma estrutura de dados chamada Pilha Dupla. Esta estrutura de dados deve oferecer a funcionalidade de duas pilhas distintas A e B (alocação sequencial) que devem ser armazenadas em um único vetor compartilhado de tamanho M . Implemente as operações de empilha/desempilha do TAD Pilha Dupla de tal forma que a soma das quantidades de elementos em A e B não extrapole M e que, tanto A quanto B possam armazenar até M elementos.

Fim da lista 2.