

- 1- Faça uso de uma pilha estática para armazenar 5 valores inteiros inseridos pelo usuário. Crie um função no main, que desempilhe todos os valores, e informe se cada valor é par ou ímpar.
- 2- Suponha que uma pilha possua 4 valores na seguinte ordem: 1, 2, 3 e 4. Qual seria a sequência correta de operações de inserção (push) e eliminação (pop) para se obter os registros na ordem 2 4 3 1?
- 3- Explique o funcionamento de uma pilha. Dê um exemplo, onde a pilha pode ser utilizada.
- 4- Suponha que uma pilha possua 4 valores na seguinte ordem: 1, 2, 3 e 4. Explique o processo de inserir e remover um elemento na pilha, apresente o passo a passo.
- 5- Considere uma pilha estática que armazene caracteres. Desenvolva um algoritmo que faça uso de uma função para determinar se uma string é da forma XY, onde X é uma cadeia formada por caracteres arbitrários e Y o reverso de X. Por exemplo, se x = ABCD, então y = DCBA. Considere que x e y são duas strings distintas. Faça uso da função `getCharAt(int indice)`, função da String que retorna um caractere a partir da posição do caractere na String.
- 6- Escreva um algoritmo que receba um número inteiro entre 0 e 16 e converta-o em binário, faça uso de uma pilha estática de 5 posições, para imprimir os valores na posição correta.
- 7- Dado uma pilha estática de 10 posições que armazene números inteiros, escreva uma função que após inserir os valores escritos pelo usuário, deve-se desempilhar em ordem crescente. Lembre-se que você pode só pode usar as funções do TAD de pilha push, pop, peek, isFull, isEmpty.
- 8- Dado uma pilha estática que armazene números, escreva uma função que forneça o maior, o menor e a média aritmética dos elementos da pilha. Faça uso apenas das funções da pilha. Após printar as informações pedidas, a pilha deve se encontrar na configuração original (os elementos na mesma ordem e posição).
- 9- Desenvolva uma função para testar se duas pilhas estáticas P1 e P2 são iguais. Duas pilhas são ditas iguais, se possui todos os elementos na mesma posição.
- 10- Dado uma pilha lida pelo usuário de 10 elementos, informar a quantidade de valores ímpares e a quantidade valores pares. Ao final a pilha deve estar na mesma configuração original. Usar apenas as operações de pilha.
- 11- Na estrutura de pilha, crie um método chamado reverse que altera a ordem dos elementos da pilha. Utilize apenas os métodos push, pop isEmpty. Não é um método para imprimir os elementos. É um método para, efetivamente, inverter a ordem dos elementos na pilha.

12- Criar um código que teste se uma fórmula é bem formada usando estrutura de pilha estática. Uma fórmula bem formada testa se para cada parêntese aberto, existe um parêntese fechado.

- Entrada: vai ser uma string inserida pelo usuário (uma fórmula)
- Saída: verdadeiro ou falso
- Exemplos:
  - String entrada: " $((9+5) + 8)$ " Saída: True
  - String entrada: " $(10*4)+((9+5))$ " Saída: false

13- Utilizando somente operações de empilhar e desempilhar, escreva um programa que remove um item com chave c (valor) fornecida pelo usuário da pilha. Ao final da execução da função, a pilha deve ser igual à original, exceto pela ausência do item removido. Note que você não tem acesso à estrutura interna da pilha (topo, item, etc), apenas às operações de manipulação.

14- Crie uma pilha estática que armazene Strings (vai ser necessário mudar o vetor de pilhas). Esta pilha irá representar um estacionamento, e deverá armazenar até 10 carros, as Strings em cada posição representa as placas de carros. Este estacionamento possui apenas uma entrada e uma saída, representado pelo TOPO. Se chegar um cliente para retirar um carro que não seja o mais próximo da saída (TOPO), todos os carros bloqueando o caminho sairão do estacionamento. O carro do cliente será manobrado para fora do estacionamento, e os outros carros voltarão a ocupar a mesma sequência inicial. Escreva um programa que processe um grupo de linhas de entrada. Cada linha de entrada contém um 'E', de entrada, ou um 'S' de saída, e a da placa do carro. Presume-se que os carros cheguem e partam na mesma ordem que entraram no estacionamento. O programa deve imprimir uma mensagem sempre que um carro chegar ou sair. Quando um carro chegar, a mensagem deve especificar se existe ou não vaga para o carro no estacionamento. Se não houver vaga, o carro partirá sem entrar no estacionamento. Quando um carro sair do estacionamento, a mensagem deverá incluir o número de carros manobrados para fora do estacionamento. Note que você não tem acesso à estrutura interna da pilha (topo, item, etc), apenas às operações de manipulação.

15- Crie um PROGRAMA que implemente o jogo Torres de Hanói. Utilize três pilhas estáticas para representar os três pinos.

Seu programa deverá obedecer a seguinte estrutura:

1. Programa inicializa o jogo com os 7 discos no primeiro pino.
  - a. Programa imprime uma visualização gráfica do estado dos três pinos apresentando onde estão os discos.
  - b. Programa lê do teclado dois números inteiros: um indicando o pino de origem e outro indicando o pino de destino. Se o pino for inválido, o usuário deverá ser avisado, e o programa deverá pedir um novo número.
2. Programa verifica se existe algum disco no pino de origem e se é possível realizar a movimentação do disco que está no topo desse pino para o pino de destino.

- a. Se existir e for possível realizar o movimento, programa deverá realizar o movimento.
  - b. não for possível, programa mostra mensagem de erro e continua no passo
3. Programa verifica se todos os discos já estão no 3º pino.
  - a. Se estiverem, programa finaliza com uma mensagem parabenizando o jogador.
  - b. Se não estiverem, programa volta a executar o passo 2.

