

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PARAÍBA Campus Campina Grande</p>	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA – CAMPUS CAMPINA GRANDE		
	CURSO:	CURSO DE ENGENHARIA DA COMPUTAÇÃO	
	PERÍODO:		TURMA:
	DISCIPLINA:	PROGRAMAÇÃO E ESTRUTURAS DE DADOS	
	PROFESSOR:	CÉSAR ROCHA VASCONCELOS	SEMESTRE LETIVO

Prática Laboratório – Pilhas Seqüenciais

- 1) Faça um programa em C que exiba o seguinte menu principal:

Editor de Pilha v1.0

=====

- 1-- Inicializar
- 2-- Empilhar
- 3-- Desempilhar
- 4-- Elemento do topo
- 5-- Imprimir pilha
- 6-- Inverter elementos da pilha
- 7-- Esvaziar a pilha
- 8-- Mudar de pilha

Digite sua opção: []

- 2) O usuário deve digitar a opção de sua preferência para executar uma operação. Utilizando técnicas de modularização de sistemas vistas em sala, implemente o tipo abstrato de dados **TPilha** (de implementação seqüencial e tipo base inteiros). Crie também uma biblioteca chamada **pilhaseq.h** contendo todos os protótipos das operações básicas vistas em sala de aula.
- Na opção 1, deve ser inicializada uma pilha vazia
 - Na opção 2, deve ser lido o elemento a ser empilhado
 - Na opção 3, deve ser desempilhado o topo. Exiba para o usuário o dado que foi desempilhado
 - Na opção 4, deve ser exibido (e não removido!) o elemento que se encontra no topo da pilha
 - Na opção 5, devem ser impressos todos os elementos da pilha.
 - Na opção 6, deve ser criada por você uma função capaz de inverter todos os elementos da pilha atual
 - Na opção 7, deve exibida uma mensagem de confirmação para o usuário antes de esvaziá-la
 - Na opção 8: permita que sejam manipuladas, no máximo, 3 pilhas simultaneamente no programa. A primeira pilha a ser utilizada pelo usuário é a pilha1. Quando a opção *mudar de pilha* for selecionada, deve-se fornecer a possibilidade do usuário manipular agora uma outra pilha de seu interesse. Assim, toda vez que o usuário desejar alternar para a pilha2 (por exemplo), todas as opções do menu devem operar sobre a pilha2. Ao mudar de pilha, preserve o estado da que estava sendo usada. Você entendeu a idéia.
 - Após a execução de cada operação, o programa deve retornar ao menu principal para que o usuário possa executar outras opções sobre a pilha corrente ou encerre o programa (no caso dele teclar 0 “zero”)
 - Lembre-se:** em cada uma das operações, identifique possíveis situações de erros do usuário e exiba mensagens para ele nestas situações. (Ex. o programa deve exibir mensagens no caso do usuário tentar desempilhar um item ou esvaziar uma pilha que está vazia, empilhar um elemento numa pilha que já está cheia, etc.). Isto é um sinal de maturidade na programação e torna seu TAD mais robusto (menos suscetível a travamentos).
- 3) Uma vez que as operações básicas foram implementadas na questão anterior, insira agora na biblioteca **pilhaseq.h** as seguintes **novas operações** (crie novos protótipos! Não use funções existentes):
- void desempilhaNvezes(TPilha *pilha, int n)** . Esta função deve receber a pilha e desempilhar N elementos a partir do topo. Cuidado! Verifique a relação entre N e o tamanho da pilha!
 - int pilhasIguais(TPilha *pilha1, TPilha *pilha2)** que possa verificar se duas pilhas são iguais em tamanho e conteúdo
- 4) Finalmente, modifique o menu de operações, inserindo todas as novas operações da questão anterior.