

	BACHARELADO EM SISTEMAS DE INFORMAÇÃO	
	Roteiro de Exercícios 7 – Pilha	
	Algoritmos e Estruturas de Dados II – 2º BSI	
Professor: Robson Alves Campêlo Entrega: 11/10		Valor: 4,0 pontos

- 1) Escreva um programa que avalia cadeias de caracteres com expressões matemáticas, que podem conter termos entre parênteses, colchetes ou chaves, ou seja, entre os caracteres '(' e ')', ou '[' e ']', ou '{' e '}'. Crie uma pilha utilizando as estruturas apresentadas na aula: `bloco.h`, `pilha.h`, `pilha.cpp` e `main.cpp`. A funcionalidade requerida da avaliação dos caracteres deverá implementada em **main.cpp**.
- 2) Escreva um programa que solicite ao usuário um texto (cadeia de caracteres) sem limite máximo de tamanho e realize as seguintes operações usando uma pilha:
 - a) Imprimir o texto na ordem inversa;
 - b) Verificar se o texto é um palíndromo, ou seja, se a string é escrita da mesma forma de frente para trás e de trás para frente. Ignore espaços, pontuações e caracteres especiais. Exemplos de palíndromos: “asa”, “ovo”, “socorram-me subi no onibus em marrocos”.
- 3) Desenvolva um algoritmo (**void inverter (pilha *P)**) para inverter a posição dos elementos de uma **pilha P**. Você pode criar pilhas auxiliares, se necessário. Mas o resultado precisa ser dado na pilha P (em uma chamada à operação `P.imprimir()`).
- 4) Desenvolva uma operação para transferir elementos de uma pilha P1 para uma pilha P2 (cópia). Siga o seguinte protótipo: **bool transferirElementos(pilha *P1, pilha *P2)** //retorna verdadeiro para sucesso e falso para erro.
- 5) Desenvolva um algoritmo para testar se duas pilhas P1 e P2 são iguais. Duas pilhas são iguais se possuem os mesmos elementos, na mesma ordem: **int iguais(pilha *p1, pilha *p2)**; //retorna 1 para `p1 == p2` e 0 para `p1 != p2`.
- 6) Em uma rua sem saída de uma cidade foi criado um estacionamento com limite máximo de 8 vagas para veículos. As vagas estão distribuídas a partir da vaga 1 (no final da rua) à vaga 8 (no início da rua). Por ser sem saída, a entrada do estacionamento é por apenas uma extremidade da rua, de modo que se um veículo quiser ocupar a vaga 3, por exemplo, os veículos das vagas 4 em diante terão que sair da rua, o que ocupava a vaga 3 é retirado para dar lugar ao novo veículo, e logo em seguida os veículos anteriores retornam ao estacionamento. Utilize uma pilha para realizar o controle desse estacionamento. Considere cada veículo como um bloco na pilha com o valor único da placa de cada veículo.
- 7) Suponha uma máquina de calcular que trabalha apenas com números não negativos, e que tem apenas quatro operações: soma, subtração, produto e divisão inteira. A máquina possui 16 teclas, representadas pelos caracteres:

0 1 2 3 4 5 6 7 8 9 + - * / C E

Onde C representa “Clear”, E representa “Enter”, que é usada para indicar que vai

ser fornecido um número. A máquina usa uma notação na qual o operador vem depois dos operandos. Escreva um programa que usa uma pilha de inteiros para simular a máquina: cada caractere que entra é tratado, e a resposta é o conteúdo final da pilha. Inicialmente, a pilha da máquina está vazia. As ações correspondentes a cada caractere são:

$i = 0, \dots, 9$	Troque o valor de x do topo da pilha por $x * 10 + i$
E	Empilhe um 0
$op = +, -, *, /$	Tire dois elementos y e x do topo da pilha, e empilhe $x \ op \ y$.
C	Esvazia a pilha.

Por exemplo, se o usuário tivesse digitado a seguinte sequência:

E 9 0 E 1 5 E 3 0 + - C

A pilha teria sucessivamente os seguintes conteúdos:

Comando	Pilha
//Começo do Programa – pilha vazia	[] <-topo
E	[0] <-topo
9	[9] <-topo
0	[90] <-topo
E	[90 0] <-topo
1	[90 1] <-topo
5	[90 15] <-topo
E	[90 15 0] <-topo
3	[90 15 3] <-topo
0	[90 15 30] <-topo
+	[90 45] <-topo
-	[45] <-topo (Resultado)
C	[] <-topo

Indicando o resultado de 45 da sequência **E 9 0 E 1 5 E 3 0 + - C**, ou seja, $(90 - (15+30))$ na notação habitual.