

Lista de Exercícios 1 – Pilhas

1. Duas pilhas A e B podem compartilhar o mesmo vetor, como esquematizado na Figura. Faça as declarações de constantes e tipos necessárias e escreva as seguintes funções:
 - 1.1. Cria pilhas, que inicia os valores de topoA e topoB;
 - 1.2. Verifica se a pilha A está vazia (vaziaA);
 - 1.3. Verifica se a pilha B está vazia (vaziaB);
 - 1.4. Verifica se as pilhas estão cheias;
 - 1.5. Empilha uma chave na pilha A;
 - 1.6. Empilha uma chave na pilha B;
 - 1.7. Desempilha A;
 - 1.8. Desempilha B.
2. Escrever um algoritmo que recebe uma sequência de caracteres e retorna a sequência na ordem inversa, usando o TAD Stack.
`char *Inverte(char *s, int n)`
3. Escreva uma nova função para o tad Stack que devolve os k elementos da pilha, ou os elementos até a pilha estar vazia.
`void *stcMultiPop (Stack *s, int k)`
4. Faça as seguintes funções para o TAD cQueue (fila em vetor circular):
 - 4.1. Que retorna o ultimo element fila (sem remover)
`void *cqGetLast(CQueue *q)`
 - 4.2. Que recebe um vetor com n elementos e os colcoa na fila.
`Int cqEnqueueN (cQueue *q, int n, void **elms)`
 - 4.3. Que devolve um vetor com os n primeiros elementos da fila.
`void *cqDequeueN (cQueue *q, int n)`
 - 4.4. Que remove e devolve um elemento identificado por uma chave na fila circular.
`void *cqDequeueSpecified (cQueue *q, void *key, int (*cmp(void *, void *))`
5. Escreva um algoritmo que usando o TAD Stack retorna TRUE se uma cadeia de caracteres consiste em um número de 'A's seguido por um número igual de 'B's. Em cada ponto, você pode ler apenas o próximo caractere da cadeia de caracteres. O algoritmo não deve contar o número de 'A's.
`int VerificaNumASeguidoB(char * str, int len)`
6. Use as operações do TAD Stack para construir operações que façam o seguinte:
 - 6.1. Dado um inteiro n, definir elm como o enésimo elemento a partir do topo da pilha, deixando a pilha sem seus n elementos superiores.
`int stkPushElmAsNRewmove(Stack *st, void *elm, int n)`
 - 6.2. Dado um inteiro n, definir elm como o enésimo elemento a partir do topo da pilha, deixando a pilha inalterada.
`int stkPushElmAsNNaoremove(Stack *st, void *elm, int n)`

6.3. Definir elm como o último elemento da pilha, deixando a pilha inalterada.(Dica: use outra pilha auxiliar.)

int stkPushElmAsUltimo(Stack *st, void *elm, int n)

7. Escreva um algoritmo para determinar se uma string de caracteres de entrada é da forma: xCy, onde x é uma string consistindo nas letras 'A' e 'B', e y é o inverso de x (isto é, se x = "ABABBA", y deve equivaler a "ABBABA"). Em cada ponto, você só poderá ler o próximo caractere da string.

int VerificaFormatoString(char *str, int n)

8. Escreva um algoritmo, usando uma Pilha, que inverte as letras de cada palavra de um texto terminado por ponto (.) preservando a ordem das palavras.

Por exemplo, dado o texto:

ESTE EXERCICIO E MUITO FACIL.

A saída deve ser:

ETSE OICICREXE E OTIUM LICAF

int InverteString(char *str, int n)

9. Escreva um programa que utilize uma pilha para verificar se uma sequência de caracteres formada somente pelos caracteres '(' e ')' (expressões aritméticas) está com a parentização correta. O programa deve verificar para ver se cada "abre parenteses" tem um "fecha parenteses" correspondente.

Ex.:

Correto: (()) (() ()) () ()

Incorreto:) (() ()) ((

int VerificaFormatoString(char *str, int n)

10. Escreva uma função que recebe uma fila armazenada em um vetor de forma circular e faz com que o primeiro elemento da fila fique na primeira posição do vetor e os demais nas posições subsequentes. Não pode usar um vetor auxiliar.

11. int cqReorganizaFila(CQueue *cq)