

Algoritmos e Estruturas de Dados II

1. Usando um dos tipos abstratos de dados estudados, faça um programa que verifique se uma entrada tem pares apropriados de parênteses, chaves e colchetes. Lembre-se que não é somente sobre a quantidade de itens abertos e fechados - depois que um elemento foi aberto, o próximo a fechar deve ser ele mesmo.

entrada: `for{int i = (j+3)*4; i < A[i]; i++}`

saída esperada: `ok`

entrada: `for{{int i = (j+3))*4; i < A[i[]; i++}`

saída esperada: `sobraram 1 {, 1), 1 [`

entrada: `while ((A[i])) i++;`

saída esperada: `) fechou, deveria ser]`

2. A notação polonesa é uma maneira de escrever expressões lógicas e aritméticas onde os operandos vêm **todos** antes da operação. $2 + 2$, por exemplo, vira $2 2 +$, e $(2 + 1) * 3$ vira $2 1 + 3 *$.

Dada uma expressão matemática terminada em `;`, escreva-a pela notação polonesa.

Considere números de 0 a 9 como operandos e `+ - * / % | &` (soma, subtração, multiplicação, divisão, módulo, ou lógico bit a bit, e lógico bit a bit) como operações.

3. Dado um baralho ordenado de n cartas, cada uma com um número de 1 a n , com a carta 1 no topo do baralho e a carta n na base, o seguinte procedimento é realizado até que haja somente uma carta no baralho:

A carta do topo é descartada e a nova carta do topo vai para a base do baralho.

Escreva um algoritmo que, dado um valor n , use um dos tipos abstratos de dados estudados para descobrir a sequência de cartas descartadas e a carta que sobra no baralho.

entrada: `5`

saída esperada: `1 3 5 4, 2`

4. Faça um algoritmo para inverter:
 - (a) uma fila de inteiros usando uma pilha;
 - (b) uma pilha de inteiros usando uma fila.
5. Organize um conjunto de operações para fazer uma lista usando apenas o tipo abstrato de dados pilha - ou seja: inserção, remoção, impressão e busca como se fosse uma lista, mas sobre uma pilha.

- (a) Qual seria a diferença para implementar uma fila da mesma maneira?
6. Dado um par de listas,
- (a) Escreva um algoritmo para gerar sua interseção
 - (b) Escreva um algoritmo para gerar sua união
7. Seja uma lista circular duplamente encadeada constituída de elementos (vetores de dois inteiros) correspondentes a instruções (descritas abaixo). Dada uma lista circular duplamente encadeada, escreva um algoritmo que percorra seus elementos, interpretando cada um como uma instrução (ver abaixo o significado de cada valor) e realizando essa instrução antes de ir até o próximo. Finalmente, o algoritmo deve realizar a impressão da lista.
- 1, i: insere o valor i na lista;
 - 2, i: remove o valor i da lista;
 - 3, i: empilha o valor i na lista (isto é, trata-a como uma pilha);
 - 4, i: desempilha a lista;
 - 5, i: enfileira o valor i na lista (isto é, trata-a como uma fila);
 - 6, i: desenfileira o valor i da lista.
- (a) Existe uma redundância no conjunto de instruções (elementos) dessa lista. Qual é?
- (b) Escreva um algoritmo que percorra os elementos de uma lista assim, interpretando-os e realizando as instruções correspondentes. Finalmente, o algoritmo deve imprimir a lista.
8. Implemente em C um algoritmo que receba uma lista de pilhas e, para cada pilha na lista, imprima o número de pilhas iguais na lista, sendo que:
- (a) Além dos elementos da entrada, seu algoritmo só pode usar **pilhas**;
 - (b) Além dos elementos da entrada, seu algoritmo só pode usar **filas**.