

Exercícios Avaliações Anteriores

Pilhas: considere a pilha implementada em sala de aula.

- utilizando somente as funções definidas, construir uma função `n_pop`, que desempilha `n` elementos da pilha. Faça todas as verificações necessárias e retorne resposta adequada ao usuário.
- utilizando somente as funções definidas, construir uma função `n_push`, que empilha o elemento recebido `n` vezes. Faça todas as verificações necessárias e retorne resposta adequada ao usuário.
- Considere agora a seguinte restrição: um valor nunca pode ser empilhado sobre um valor menor que ele, por exemplo

2	É uma pilha válida	5	NÃO é uma pilha válida
5		6	
9		8	
12		7	

Escrever uma função que tem como entradas uma pilha `p` de acordo com a restrição e um elemento `n`. A sua função deve empilhar o elemento `n` em `p`, sem quebrar a restrição, utilizando as funções comuns de pilha. Qualquer validação que considere necessária deverá ser feita na sua função. Você pode usar pilhas auxiliares se achar necessário. O empilhamento é obrigatório, no entanto, o lugar de empilhamento deve obedecer à restrição. Utilize as funções de manipulação.

- Construir uma função que recebe uma pilha `p` e um inteiro `n`, remove todas as ocorrências desse inteiro, se ele estiver na pilha. A função devolve 1 em caso de sucesso ou zero em caso de fracasso (não existir o valor `n` na pilha). Não se esqueça de preservar a pilha, a menos dos valores retirados. Você pode usar pilhas auxiliares na sua função. Utilize as funções de manipulação.

Recursão:

- Considere o vetor `v = {3, 5, 2, 6, 1}` e a função a seguir

```
int rec (int v[], int i) {
    int x;
    if (i == 0)
        return v[0];
    x = rec (v, i-1);
    if (x > v[i])
        return x;
    return v[i];
}
```

- Execute o teste de mesa dessa função para chamada `printf ("\n%d\n", rec(v, 4));`
- O que será impresso? O que significa esse valor?

- Escrever uma função recursiva para somar os inteiros de 1 a `n`.

- Considere o código a seguir, execute um teste de mesa para o programa e diga o que este programa faz.

```
#include <stdio.h>

int zum (char c, char s[], int i) {
    if (s[i] == '\0')
        return 0;
    if (s[i]==c) return (1+zum(c, s, i+1));
    return zum(c, s, i+1);
}

int main(void) {
    char *s="teste", c='t';
    int t=zum(c, s, 0);
    printf("\n\nEncontrei %d vezes", t);
    return 0;
}
```