



Exercícios – Pilhas

Orientações:

- Apenas os exercícios marcados com um * deverão ser entregues.
- Os exercícios poderão ser realizados de forma individual ou em dupla.
- Data limite de entrega: **04/04 às 23h59** no escaninho de apenas um dos alunos da dupla.
- Organizar todas as respostas em um arquivo único .pdf contendo o nome e RA dos dois alunos. O nome do arquivo .pdf deve ser: **exercicio3_RA1_RA2.pdf**
- Arquivos fora do prazo, com formato e nome diferentes do especificado não serão considerados.
- Lembrem-se da nossa regra sobre cópia de respostas! Em caso de dúvidas, consulte a professora.

1. * Implementar a função `size()` que retorna o total de elementos na pilha.
2. * Implementar a função `showStack()` que exibe todos os elementos da pilha do topo até a base. A pilha original deve ser mantida.
3. Suponha que objetos 1,2,3,4 são colocados, nesta ordem, numa pilha inicialmente vazia. Depois de empilhar um objeto, você pode tirar zero ou mais elementos da pilha. Faça uma função onde cada elemento desempilhado é exibido. Por exemplo, a sequência de operações empilha 1, empilha 2, desempilha, empilha 3, desempilha, desempilha, empilha 4, desempilha, produz a impressão da sequência 2,3,1,4.
4. * Considere uma *string* `s` contendo apenas os caracteres: '+' e '-'. Onde o caractere '+' representa a operação *push()* e '-' a operação *pop()*. Construir uma função que recebe uma *string* `s`, como definido, e retorna -1 se a *string* gera um *underflow* (mais operações de pop), retorna 0 se gera um *overflow* (mais operações de push) e 1 se *ok*. Considere ainda que a pilha tem limite para no máximo `n` elementos.
5. Construir uma função que cria uma pilha para conter elementos da seguinte forma: 1º elemento (base): 1!; 2º elemento: 2!, e assim por diante, até `n!`, onde `n` é um número inteiro, lido no *main* e passado como parâmetro para a função.
6. Escreva um programa para determinar se uma *string* de caracteres de entrada é da forma: `x y` 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*.



7. Dadas duas pilhas p1 e p2, ambas contendo números inteiros quaisquer. Considere que as pilhas possuem o mesmo número de elementos. Construir uma função que monta uma terceira pilha p3, para conter em cada posição um número da sequência de Ackermann:

$$A(m, n) = \begin{cases} n + 1, & \text{se } m = 0 \\ A(m - 1, 1), & \text{se } n = 0 \\ A(m - 1, A(m, n - 1)), & \text{caso contrário} \end{cases}$$

onde, m representa cada elemento da pilha p1 e n da pilha p2.
No final as pilhas p1 e p2 deverão estar restauradas.

8. * Brinde Face 2015:
<https://www.urionlinejudge.com.br/judge/pt/problems/view/1944>
9. * Estacionamento Linear:
<https://www.urionlinejudge.com.br/judge/pt/problems/view/1523>
10. Ácido Ribonucleico Alienígena:
<https://www.urionlinejudge.com.br/judge/pt/problems/view/1242>