

## Lista de Exercícios sobre pilhas

1. Construir uma função que remove  $n$  elementos de uma pilha. Verificar os parâmetros e fazer as validações necessárias dentro da função. Utilizar somente as funções definidas em `pilha.h`
2. Construir uma função que recebe uma pilha e a devolve invertida.
3. Construir uma função que recebe uma pilha `p1` e devolve outras duas:
  - a. Uma com os pares de `p1`,
  - b. Outra com os ímpares de `p1`;
4. Duas pilhas sequenciais numéricas estão ordenadas crescentemente a partir o topo. Construir uma função que transfere os elementos dessas pilhas para uma terceira inicialmente vazia, de modo que ela fique ordenada decrescentemente, ou seja, com o maior valor no topo.
5. Criar uma estrutura pilha para armazenar caracteres e definir todas as funções necessárias para sua manipulação.
  - a. Escrever uma função para essa pilha de caracteres que verifica se expressões aritméticas estão com a parentização correta. Dica: os números serão todos de um único algarismo. Sua função deve checar expressões para ver se cada "abre parênteses" tem um "fecha parênteses" correspondente. O protótipo da função é:

```
int parenteses_corretos(char *);
```

A função retorna 1 se expressão tiver com a parentização correta ou 0, caso contrário.

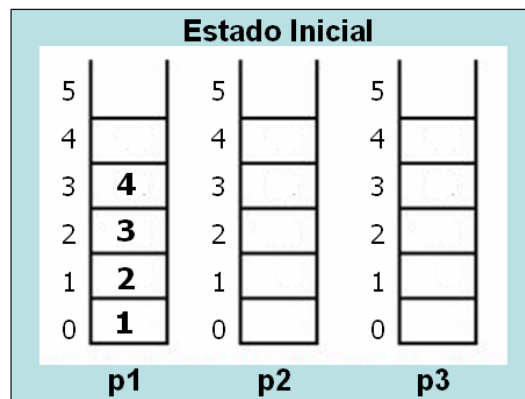
Exemplos

- $2+(5*4]$  não está correta
- $3- 4 +7)$  não está correta
- $2- \{ 4*5$  não está correta
- $4+((3-5\{\}))$  não está correta

- b. Escrever uma função para verificar se uma frase é palíndromo. Dica: eliminar os espaços antes.
6. A conversão de um valor decimal para o seu correspondente em binário é feita por sucessivas divisões desse número por 2 até que o quociente seja 0. O representante binário desse número será composto por todos os restos das divisões, só que na ordem inversa à que foram calculados. Elabore um algoritmo para resolver essa questão, utilizando o conceito de pilhas e depois fazer uma implementação na linguagem C. O número é fornecido pelo usuário.

$\begin{array}{r} 13 \overline{) 2} \\ -12 \phantom{0} \\ \hline 1 \end{array}$	$\begin{array}{r} 6 \overline{) 2} \\ -6 \phantom{0} \\ \hline 0 \end{array}$	$\begin{array}{r} 3 \overline{) 2} \\ -2 \phantom{0} \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \overline{) 2} \\ -0 \phantom{0} \\ \hline 1 \end{array}$
$\longleftarrow$			

7. Dado o estado inicial das pilhas p1, p2 e p3 na figura a seguir, mostre por meio de desenho, o estado final delas após as operações descritas no código a seguir. Considere que p1, p2 e p3 sejam variáveis do tipo `t_pilha` visto em aula.



```
int temp = desempilha (&p1);
empilha (temp, &p2);
empilha (desempilha(&p1), &p3);
empilha (desempilha(&p1), &p2);
temp = desempilha (&p1);
empilha (temp, &p3);
empilha (desempilha(&p2), &p1);
empilha (desempilha(&p2), &p3);
empilha (desempilha(&p1), &p3);
```

8. Construir uma função que transfere os dados de uma pilha p1 para uma pilha p2, utilizando as funções de manipulação de pilhas, de maneira que eles mantenham a ordem original. Sugestão: Use uma pilha auxiliar.
9. Escrever uma função para verificar se um dado item está presente em uma pilha. Em caso positivo, a função devolve a posição do item na pilha, considerando o topo como posição 1. Caso o elemento não esteja presente, a função devolve -1. A pilha deve permanecer a mesma após a execução do procedimento.