

## Exercícios: revisão para AP1

Para as questões que utilizarem implementações já feitas em sala de aula você poderá usar os arquivos da pasta no Google Drive (link).

### Questão 1.

Acrescente a função `int List_count(List* l)` ao arquivo `List.h` que implementa uma lista de inteiros. A função precisa usar recursividade para contar o número de elementos na lista encadeada.

### Questão 2.

Acrescente a função `int List_invert(List* l)` ao arquivo `List.h` que implementa uma lista de inteiros. A função precisa usar recursividade para inverter a ordem dos elementos na lista encadeada.

### Questão 3.

Acrescente a função `List *List_concat(List* l1, List* l2)` ao arquivo `List.h` que implementa uma lista de inteiros. A função deve retornar uma nova lista com os elementos das listas 1 e 2 concatenados. Exemplo:  $l_1 = \{1, 3, 5\}$ ,  $l_2 = \{2, 4, 6\}$  a lista resultante será  $l_3 = \{1, 3, 5, 2, 4, 6\}$ .

### Questão 4.

Use as operações/funções Empilha (push), Desempilha (pop) e Vazia (isEmpty) para construir operações que façam o seguinte:

- Definir o item  $i$  como o segundo elemento do topo da pilha.
- Definir o item  $i$  como o último elemento a partir do topo da pilha.
- Dado um inteiro  $n$ , definir o item  $i$  como o  $n$ -ésimo elemento a partir do topo

### Questão 5.

Para um dado número inteiro  $n > 1$ , o menor inteiro  $d > 1$  que divide  $n$  é chamado de fator primo. É possível determinar a fatoração prima de  $n$  achando-se o fator primo  $d$  e substituindo  $n$  pelo quociente  $n/d$ , repetindo essa operação até que  $n$  seja igual a 1. Utilizando um dos TADs vistos em sala (Lista, Pilha ou Fila) para auxiliá-lo na manipulação de dados, implemente um programa que compute a fatoração prima de um número imprimindo os seus fatores em ordem decrescente. Por exemplo, para  $n = 3960$ , deverá ser impresso  $11 * 5 * 3 * 3 * 2 * 2 * 2$ . Justifique a escolha do TAD utilizado.

### Questão 6.

Considero que um estacionamento da Rua Direita, em Ouro Preto, é composto por uma única alameda que guarda até dez carros. Existe apenas uma entrada/saída no estacionamento, e esta extremidade da alameda dá acesso justamente à Rua Direita. Se chegar um cliente para retirar um carro que não seja o mais próximo da saída, todos os carros bloqueando seu caminho sairão do estacionamento. O carro do cliente será manobrado para fora do estacionamento, e os outros carros voltarão a ocupar a mesma sequência inicial.

Escreva um programa que processe um grupo de linhas de entrada. Cada linha de entrada contém um 'E', de entrada, ou um 'S' de saída, e o número da placa do carro. Presume-se que os carros cheguem e partam na mesma ordem que entraram no estacionamento. O programa deve imprimir uma mensagem sempre que um carro chegar ou sair. Quando um carro chegar, a mensagem deve especificar se existe ou não vaga para o carro no estacionamento. Se não houver vaga, o carro partirá sem entrar no estacionamento. Quando um carro sair do estacionamento, a mensagem deverá incluir

o número de vezes em que o carro foi manobrado para fora do estacionamento para permitir que os outros carros saíssem.

**Questão 7.**

Considere uma pilha  $P$  vazia e uma fila  $F$  não vazia, ambas de inteiros. Utilizando apenas os testes de fila e pilha vazias, as operações da fila e da pilha para enfileirar, desenfileirar, empilhar e desempilhar, mais uma variável auxiliar inteira, escreva uma função que inverta a ordem dos elementos da fila.

**Questão 8.**

Implemente uma fila em  $C$ , onde cada item da fila consista em um número variável de inteiros.