

Questão 1.

3 P.

Seu Dionízio tem uma pastelaria e precisa organizar os pedidos que chegam na cozinha para que os clientes não saiam insatisfeitos com o atendimento. Ele então resolveu recorrer a um sistema que permita a ele gerenciar a ordem dos pedidos que entram e saem da cozinha, além de calcular o valor dos pedidos das mesas para que nenhum garçom erre as contas.

A pastelaria só faz dois sabores de pastel, carne ou queijo (R\$ 3,00). Além do pastel, os pedidos ainda podem conter um refrigerante (R\$ 2,00).

1.0P Crie uma estrutura **Pedido** que armazene o número do pedido, quantos pastéis (de cada sabor) e quantos refrigerantes cada cliente pediu.

1.0P Crie um programa em C para realizar pedidos. O programa deve apresentar duas opções: (1) Fazer um pedido e (2) Listar pedidos. Enquanto o usuário selecionar a opção 1 o programa deve perguntar qual o número de pastéis de carne, de queijo e quantos refrigerantes. Os pedidos devem ser armazenados em uma fila do tipo **Pedido**.

1.0P Quando a opção 2 for selecionada o programa deve desenfileirar os pedidos e mostrar um relatório com as informações de cada pedido, mais o seu total.

Pedido	Carne	Queijo	Refri	Total
1	2	2	4	R\$ 20,00
2	2	0	2	R\$ 9,00

Questão 2.

2 P.

Considere uma lista de valores inteiros e implemente a função `void List_unique(List *l)` que recebe como parâmetro uma lista e remove todas as ocorrências duplicadas.

Exemplo: $[5] \rightarrow [8] \rightarrow [3] \rightarrow [5] \rightarrow [8] \rightarrow [2] \Rightarrow [5] \rightarrow [8] \rightarrow [3] \rightarrow [2]$

Questão 3.

3 P.

1.5P Crie a função `void *List_getValue(List *l, int index)`, que devolve (sem remover) o n-ésimo item de uma lista genérica lista l.

1.5P Crie a função `int List_getIndex(List *l, void *value, int (*cmp)(void*,void*))`, que devolve a posição (índice) da primeira ocorrência do valor v em uma lista genérica.

Questão 4.

2 P.

Crie um programa em C que implemente um algoritmo usando a estrutura de dados pilha para a verificação de parênteses desequilibrados.

O programa deve receber uma expressão formada por números, operadores básicos (+, -, *, /) e parênteses pelo console e informar se os parênteses da expressão estão desbalanceados ou não.

Uma abordagem para verificar parênteses equilibrados é usando uma pilha. Cada vez que um parêntese aberto for encontrado, coloque-o na pilha, e quando um parêntese de fechamento for encontrado remova o topo da pilha. Se no final a pilha estiver vazia, não há desbalanceamento, caso contrário há.

Exemplos:

$2 * 14 * (3 + 15) / 4 - (9 * 2 \rightarrow$ Desbalanceada
 $(23 - 10) + 37 / (100 * 14) - 1 \rightarrow$ Balanceada