
	<p style="text-align: center;">MINISTÉRIO DA EDUCAÇÃO UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB CURSO DE SISTEMAS DE INFORMAÇÃO Disciplina: Estrutura de Dados I Professora: Luana Batista da Cruz</p>	
---	--	---

TERCEIRA AVALIAÇÃO (EM DUPLA) – VALE 6 PONTOS

1. O estacionamento Luzeiros contém uma única alameda que guarda até 10 carros. Existe uma entrada e uma saída, de tal forma que quando um determinado carro entra ele fica no final da fila e o primeiro que chegou sempre fica próximo a saída, ou seja, todas as vezes que um carro sai todos os outros devem ser manobrados um espaço para frente. Faça um programa em C, onde o usuário entrará com o número da placa do carro e 'E' se estiver entrando no estacionamento e 'S' se estiver saindo do estacionamento. O programa deve emitir uma mensagem sempre que um carro entrar ou sair do estacionamento. Quando um carro chegar, a mensagem deve especificar se existe ou não vaga no estacionamento. Quando um carro sair, a mensagem deverá incluir o número de vezes em que o carro foi manobrado para fora do estacionamento para permitir que outros carros saíssem, além de mostrar quantos carros foram manobrados para que ele saísse (1,0 ponto).

Obs1: Não se esqueça que todas as vezes que um carro vai sair os que estão na frente devem ser manobrados para o final da fila, mas ao final o carro que estava no início da fila deve continuar como o primeiro da fila.

Obs2: Faça um menu de opções.

2. Refaça a questão 1 substituindo fila estática por fila dinâmica (1,0 ponto).
3. Faça um programa em C onde o usuário digita uma expressão matemática no modo in-fixa e então o programa verifica se a expressão é válida, depois use pilha estática para converter para o modo pós-fixa (1,0 ponto).

Obs1: A expressão deve ser lida em uma string, e a string deve ser no máximo de 100 caracteres.

Obs2: A expressão só pode conter dígitos e os operadores: (+, -, *, /).

Obs3: Cada número e operador deve ser separado por um espaço em branco, podendo assim um número possuir mais de um dígito. Exemplo: 130 + 50 - 25

Obs4: A expressão **DEVE** sempre iniciar com abre parênteses e terminar com fecha parênteses. Exemplo: ((130 + 50) – 25).

4. Faça um programa em C onde o usuário digita uma expressão matemática no modo pós-fixa e verifica se a expressão é válida. Posteriormente, retorne o resultado da expressão. Exemplo: ((23 2 +) 2 *). Saída = 50 (1,0 ponto).

Obs.: Use pilha dinâmica

5. Faça um programa em C que crie uma lista duplamente encadeada de Pessoas ordenada pelo CPF. Para cada pessoa, o usuário deve informar o CPF, nome e a idade, e depois faça as seguintes funções que (1,0 ponto):
 - a) Devolva uma lista ordenada que tenha somente as pessoas menores do que uma idade informada pelo usuário.
 - b) Dado o CPF de uma pessoa devolva a idade do vizinho mais velho desta pessoa e devolva também se ele é mais velho do que a pessoa informada.
6. Faça um programa em C que leia uma fila de inteiros e uma pilha de inteiros depois devolva uma pilha contendo os elementos intercalados da fila e da pilha de tal forma que o elemento removido da fila será empilhado primeiro, desde que seja par, caso contrário empilhe primeiro o elemento desempilhado da pilha (1,0 ponto).

Obs.: Use abordagem dinâmica.

Equipe: os programas podem ser feitos em dupla, mas os relatórios são individuais (**relatórios iguais, as questões são anuladas**). A dupla deve ser identificada no envio do código.

Data de entrega: 22/11.

Entregar: código fonte + relatório (comentários da lógica do funcionamento das questões de forma detalhada).

Entrevista individual: agendar horário na [planilha disponibilizada](#).