



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
Rua Dom Manoel de Medeiros, s/n – Dois Irmãos 52171-900 Recife-PE
Fone: 81 3302 1000 www.dc.ufrpe.br

DISCIPLINA: Algoritmos e Estruturas de Dados	CÓDIGO: 06214
DEPARTAMENTO: Computação	ÁREA: Informática
CURSO: Licenciatura em Computação	
PROFESSOR RESPONSÁVEL: Luciano Demétrio Santos Pacífico	
DATA MÁXIMA DE ENTREGA: 12-09-2020	

Regras da Lista de Exercícios 04

1. **Não é permitido o uso de Estruturas de Dados prontas de Linguagens de Programação.** O aluno deve implementar suas próprias Estruturas de Dados. Na Linguagem de Programação **C**, deve-se usar **structs**. Nas demais Linguagens de Programação permitidas (vide **Regras da Disciplina**), deve-se usar **classes**.
2. **Não é permitido o uso de Algoritmos e comandos otimizados prontos de Linguagens de Programação.** Todos os algoritmos solicitados devem ser implementados pelos alunos como **procedimentos** (funções, métodos, etc.).
3. As questões que solicitam **escrita de código** devem ser resolvidas **apenas através dos recursos oferecidos pela pseudolingagem definida para a disciplina, e dos recursos equivalentes em Linguagens de Programação reais**, sendo eles: variáveis, constantes e tipos primitivos, expressões, estruturas condicionais, estruturas de repetição, sub-rotinas, estruturas de dados homogêneas (Arrays) e estruturas de dados heterogêneas (registros – classes e structs).
4. Para a Lista de Exercícios 04, **todas as Estruturas de Dados devem ser Encadeadas (Dinâmicas)**, não sendo permitida a alocação de memória de forma **Sequencial (Estática)**. As Estruturas de Dados devem ser implementadas pelos usuários, não sendo permitido o uso de **Estruturas de Dados Dinâmicas** implementadas em Linguagens de Programação, como os **Vectors** e **Lists** da Linguagem de Programação **Java**, ou **Lists** da Linguagem **Python**, por exemplo.
5. **Regra de Ouro:** Todos os alunos envolvidos em **cópias** terão suas notas **ANULADAS** nas referidas questões.
6. Apenas o código “.c”, “.cpp”, “.java”, “.py”, etc. deve ser enviado ao professor para cada questão. Deve-se enviar **um único arquivo resposta por questão**, que conterá todas as classes/estruturas e procedimentos necessários para a solução da questão. Todos os arquivos devem ser enviados **em uma única pasta, “zipados”**.
7. O arquivo de resposta com o código para cada questão deve ser **nomeado** na forma “L#Q%.c”, “L#Q%.java”, etc., onde “#” refere-se ao número da lista e “%” refere-se ao número da questão (ex.: L1Q2.c, para o arquivo de resposta da segunda questão da primeira lista, usando a Linguagem C).
8. As respostas da Lista de Exercícios 04 devem ser submetidas **unicamente através da tarefa criada no Google Classroom para este propósito**.

Lista de Exercícios 04 – Listas Lineares Encadeadas

1. Uma cadeia de caracteres é considerada “bem formada” se parênteses, chaves e colchetes são fechados corretamente. **Isso ocorre apenas quando a ordem de fechamentos desses elementos for inversa à ordem de abertura dos mesmos.** Implemente, em uma **Linguagem de Programação**, as Estruturas de Dados básicas e os Algoritmos necessários para avaliar se uma cadeia de caracteres passada como entrada, é “bem formada” ou “malformada”, de acordo com os critérios acima. Verifique os arquivos de entrada e saída exemplos (L4Q1.in e L4Q1.out, respectivamente). (4.0 pontos)

2. Escreva os algoritmos, usando uma **Linguagem de Programação**, de **Visualização (Impressão)**, **Busca**, **Inserção** e **Remoção** em uma **Lista Linear Duplamente Encadeada Circular Ordenada**, com ponteiros para o **primeiro** e **último** elementos: (6.0 pontos)
- a. Defina as Estruturas de Dados básicas necessárias.
 - b. A lista deverá estar inicialmente vazia.
 - c. A lista poderá ser **impressa** tanto de forma **direta** (do primeiro ao último elemento), quanto em forma **reversa** (do último ao primeiro elemento).
 - d. A lista deverá se manter ordenada após as operações de **inserção** e **remoção**.
 - e. Deve haver uma forma de **interação do usuário** com o programa, permitindo que esse usuário execute as operações de visualização, busca, inserção e remoção, **de acordo com a sua necessidade**.
 - f. Os procedimentos devem imprimir **mensagens de erro**, caso o usuário tente realizar **operações inválidas** (com remoções de dados inexistentes na lista, ou inserção de mais elementos do que o suportado pela Estrutura de Dados).
 - g. Não deve ser permitida a **inserção de elementos com chaves iguais** na estrutura, havendo necessidade de **impressão de uma mensagem de erro**, caso o usuário tente realizar esse tipo de inserção.