

UENF

Universidade Estadual do Norte Fluminense Darcy Ribeiro

Curso: Ciência de Computação **Data:** 04/04/2023 **Atividade:** LE1

Período: 3º

Disciplina: Estrutura de Dados I

Professor: Fermín Alfredo Tang **Turno:** Diurno

Nome do aluno: Frederico Rangel Sader e José Arthur de Mello Mulatinho

Matrícula:

Questões da LE1

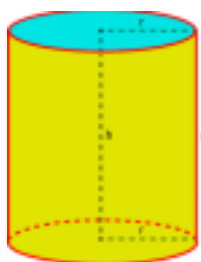
1.- (2,0 Pontos) Considerando questões de alocação de memória e estruturas de dados, tomando como base o capítulo 2, do livro: “Estrutura de Dados com Algoritmos e C - Marcos Laureano”, realize as seguintes tarefas:

- i) Implementar os programas 2.4, 2.5, 2.7 e 2.8, corrigindo caso necessário;
- ii) Explique a diferença entre o programa 2.7 e 2.8 no tratamento das matrizes.

No programa 2.8, a função "aloca" recebe o endereço de um ponteiro duplo e passa como um ponteiro triplo, alocando nesse endereço um bloco de memória com a matriz funcionando assim como no 2.7, um vetor de ponteiros para inteiros. Já no 2.7, a função "aloca" retorna o ponteiro duplo apontando para esse vetor de ponteiros para inteiros.

2.- (2,0 Pontos) Considerando os conceitos sobre TADs, consulte o capítulo 4 do livro: Estruturas de Dados Descomplicada – Andrés Backes, e realize as seguintes tarefas:

- i) Implementar um TAD que represente um cilindro.
 - a) Defina a estrutura para armazenar os dados do cilindro (posição, raio e altura);
 - b) Defina as funções para criar/inicializar uma instância de cilindro; alterar os dados do cilindro; funções que retornem a sua área e o seu volume; e uma função que destrua uma instância do cilindro.
- ii) Teste o seu TAD com uma aplicação de exemplo, que permita criar pelo menos dois cilindros, modificar seus dados, e calcular seus volumes e áreas.



3.- (2,0 Pontos) Considerando os conceitos sobre eficiência de algoritmos, consulte o capítulo 3 de “Estruturas de Dados Descomplicada - Andrés Backes”, e realize as tarefas:

- Implementar os algoritmos de ordenação: BubbleSort, SelectionSort e InsertionSort e teste cada algoritmo usando um vetor de tamanho $n=10$.
- Gerar um vetor aleatório de tamanho n , para n suficientemente grande, (ver a sugestão dada na Tabela);
- Realizar um experimento, executando os três algoritmos usando o mesmo vetor aleatório gerado em ii), e registre o tempo de ordenação de cada um (em milissegundos);
- Repetir o experimento, modificando os algoritmos de forma a contar o número de operações realizadas (comparações e atribuições) por cada algoritmo. Considere que cada troca equivale a 3 atribuições, e que um deslocamento equivale a uma atribuição.

Tam. n	Bubble Sort			Selection Sort			Insertion Sort		
	<i>Tempo (ms)</i>	<i>Nº Comp</i>	<i>Nº Atrib</i>	<i>Tempo (ms)</i>	<i>Nº Comp</i>	<i>Nº Atrib.</i>	<i>Tempo (ms)</i>	<i>Nº Comp</i>	<i>Nº Atrib</i>
50.000	6649	1249812265	1861566168	2508	1249975000	406334	1662	618916028	617161262

4.- (2,0 Pontos) Considerando os conceitos sobre eficiência de algoritmos, realize as seguintes tarefas:

- Implementar as versões: A) iterativa e B) recursiva do algoritmo para calcular o n ésimio número da série de Fibonacci. Teste cada algoritmo usando parâmetro $n=5$;
- Realizar um experimento, executando os dois algoritmos, para n suficientemente grande, de forma a identificar o limite de execução de cada algoritmo. Registre o tamanho do problema e o tempo de execução em cada caso, conforme a tabela.

Fibonacci Iterativo		Fibonacci Recursivo	
Tam. n	<i>Tempo (seg.)</i>	Tam. n	<i>Tempo (seg.)</i>
10	0.02034	10	0.02124
20	0.01898	20	0.006185
30	0.01994	30	0.1182
40	0.02130	40	0.5536
50	0.06764	50	66.93
100	0.02041		

5.- (2,0 Pontos) Escreva um **algoritmo recursivo** para determinar se uma *string* é um palíndromo. Uma *string* é um palíndromo se pode ser lida para frente e para trás com o mesmo significado. As letras maiúsculas e os espaços podem ser ignorados. Teste seu algoritmo com os seguintes palíndromos:

- A base do teto desaba.
- A cara rajada da jararaca.
- Acuda cadela da Leda caduca.
- A dama admirou o rim da amada.
- A droga do dote é todo da gorda.

E como pelo menos um caso que não seja palíndromo.

Atribuído: Em Duplas.

Entrega: Código Fonte

Apresentar: Execução e Explicação