

Curso: Análise e Desenvolvimento de Sistemas		Série: 3S	Turma: C	Turno: Noturno
Professor(a): João Paulo Biazotto		Horário: 19h às 22h30		
Acadêmico (a):				RA:
Disciplina: Estrutura de Dados				Data:
Prova	Prova Prática	Atividades de estudo programadas (AEP)	Prova integrada	Nota final do bimestre
5,0	3,0	1,0	1,0	

INSTRUÇÕES PARA REALIZAÇÃO DA PROVA:

- ⇒ Os dados do cabeçalho deverão ser preenchidos com letra maiúscula. E as questões deverão ser respondidas com letra legível.
- ⇒ É vedado, durante a prova, o porte e/ou o uso de aparelhos sonoros, fonográficos, de comunicação ou de registro eletrônico ou não, tais como: notebooks, celulares, tablets e similares.
- ⇒ A prova é individual e sem consulta, deverá ser respondida a caneta azul ou preta. Prova escrita a lápis não dá direito à revisão. Não é permitido o uso de corretivo.
- ⇒ É obrigatória a permanência do acadêmico 1 (uma) hora em sala de aula após o início da prova.
- ⇒ Não será permitida a entrada na sala de aula após 10 minutos do início da prova.
- ⇒ É obrigatória a assinatura da lista de presença impressa na qual constam RA, nome e curso.
- ⇒ O valor de cada questão está ao lado da mesma.
- ⇒ Todas as respostas devem constar no espaço destinado e autorizado pelo professor, à resposta.
- ⇒ Em caso de qualquer irregularidade comunicar ao Professor ou fiscal de sala.
- ⇒ Ao término da prova, levante o braço e aguarde atendimento do professor ou do fiscal.

1º bimestre	2º bimestre	1ª substitutiva	2ª substitutiva
--------------------	--------------------	------------------------	------------------------

GABARITO OBJETIVAS

Questão	Resposta
8	
9	
10	

1. **(0,6 ponto)** Segundo Drozdek (2002), os ponteiros são um conceito central em C e têm um papel crucial na manipulação eficiente de dados e na implementação de estruturas de dinâmicas de dados, como listas encadeadas, árvores e grafos. Os ponteiros permitem que um programa acesse, modifique e gerencie os dados de forma eficaz, referenciando sua localização na memória.

DROZDEK, Adam. **Estrutura de Dados e Algoritmos em C++**.
Cengage, 2002.

Considere que você está desenvolvendo um programa em linguagem C para manipular matrizes. No entanto, o tamanho das matrizes não é conhecido em tempo de construção e deve ser alocado em tempo de execução. Sua tarefa é implementar uma função que realize a alocação dinâmica de memória para uma matriz.

COMPLETE a implementação da função **alocarMatriz** que recebe o número de linhas e colunas como parâmetros e retorna um ponteiro para uma matriz de inteiros alocadas dinamicamente. Obs. i e j são variáveis globais do tipo int e não se preocupe em verificar se há memória ou não disponível para alocação.

```
int** alocaMatriz(int linhas, int colunas) {  
    int** matriz = malloc(linhas * sizeof(int*));  
    for (i = 0; i < linhas; i++) {
```

```
}  
    return matriz;  
}
```