


## AVALIAÇÃO P1

	CURSO: SISTEMAS DE INFORMAÇÃO	PERÍODO: 2 º
	DISCIPLINA: Algoritmo e Estrutura de Dados 1	
Avaliação Formal (0 -10):		NOTA FINAL:
Avaliação Processual (0 -10):		
ALUNO (A): Matheus Patrick Aires Lopes		

### LEIA COM ATENÇÃO AS INSTRUÇÕES ABAIXO

#### ORIENTAÇÕES GERAIS

1. Preencha seu nome completo no local indicado para Aluno (a)
2. Tempo de duração: 48 horas
3. Durante a realização da avaliação, o professor não poderá avaliar para o aluno se sua resposta está certa ou errada, nem se está completa ou incompleta.
4. **Questões sem desenvolvimento serão consideradas erradas.**
5. **Esta prova é individual e sem consulta;**
6. **A capacidade de interpretação das questões faz parte da avaliação;**

#### ORIENTAÇÕES PARA APLICAÇÃO DA PROVA ONLINE

1. Após a finalização da prova, ela deverá ser encaminhada utilizando o moodle. Caso o moodle ou a Internet apresente alguma instabilidade, a prova poderá ser encaminhada por e-mail no prazo e horário limite do término. O assunto no e-mail deverá ser: P1 – NOME DA DISCIPLINA
2. Questões que envolvem códigos, podem ser desenvolvidas em uma IDE, sugestão que crie-se um pacote para cada exercício ao final comprimir: os arquivos java, os pacotes e as questões da prova discursivas em um único arquivo zip. E anexar no Moodle.
3. Tenha preferência por responder as questões discursivas ou dissertativas e entregar a prova no arquivo formato para Word: doc ou docx. Não gerar PDF.

**Nas questões objetivas, apresente a opção correta e justifique a opção escolhida.**

**QUESTÃO 1 (Valor da questão: 2.5)** – Construa um algoritmo para o funcionamento de uma agenda. Devem ser lidos os seguintes dados de dez pessoas: nome, endereço, CEP, bairro e

telefone. Tais dados devem ser armazenados na agenda, cuja representação é uma matriz com 10 linhas (referentes às pessoas) e 5 colunas (referentes aos dados). Crie as entradas no qual os dados são pedidos ao usuário, e estes dados vão sendo armazenados na matriz. Crie uma função que receba como parâmetro uma matriz e um nome lido pelo usuário, deve imprimir os dados da pessoa referente ao nome, se o nome não existir, deve informar: "Pessoa não encontrada".

**QUESTÃO 2 (Valor da questão: 2.0)** – Crie um programa em Java, que leia um vetor de inteiros com 150 posições gerados randomicamente, os valores randômicos não podem ser maiores que 250. Faça o que se pede:

- Uma função booleana, que receba como parâmetro um vetor e um número lido pelo usuário, retorne verdadeiro se o número estiver no vetor, false se o número não existir no vetor.
- Uma função com o maior elemento do vetor.
- Uma função com o menor elemento do vetor.
- A média aritmética dos elementos do vetor.

**QUESTÃO 3 (Valor da questão: 3.5)** – Crie um algoritmo que simule um jogo de bingo. Faça da seguinte forma: crie 5 matrizes que vão ser chamadas da seguinte forma: jogador1, jogador2, jogador3, jogador4, jogador5. Todas matrizes são 5x5, contém números randômicos (aleatórios) entre 0 e 99 não repetidos (os números não se repetem dentro da matriz, mas podem existir matrizes com números iguais). A cada rodada um número é gerado aleatório, este número é verificado em cada cartela de bingo (matriz), quando encontrado, esta posição é marcada. O jogo acaba quando uma linha ou coluna inteira de qualquer matriz for preenchida, o jogo deve anunciar que o jogo acabou, e existe um vencedor, apresentar o vencedor.

**Obs.:** O raciocínio criado para definir como será marcado o número da matriz faz parte do exercício, e deve ser explicado como comentário no código do exercício.

**QUESTÃO 4 (Valor da questão: 2.0)** – A partir de uma pilha sequencial e estática, contendo 26 elementos empilhados, faça o que se pede:

- crie uma função que encontre o maior elemento da pilha e coloque-o no topo
- crie uma função com retorno inteiro, receba a pilha e um elemento por parâmetro, a função deve retornar a quantidade de elementos que foram necessários desempilhar para encontrar o elemento. Ao final da função apenas o elemento passado por parâmetro deve ser retirado, os demais elementos devem permanecer na pilha como no original, se o elemento não for encontrado retornar -1, se o elemento for o topo retornar 0.

Obs.: Use as operações de empilha e desempilha, as demais características da pilha: vetor, topo não são acessíveis no main. O código a ser entregue, deve conter a estrutura de pilha sequencial estática, e o desenvolvimento do exercício pedido.

**QUESTÃO 5 (Valor da questão: 0.5)** – Exercício extra em sala.