

		<b>Prova de Fundamentos de Programação</b>	
		<b>Professor:</b> Gleison Brito	
<b>Turma:</b> BHZ231BMTDS		<b>Data:</b>	
<b>Aluno(a):</b>		<b>Número:</b>	
<i>As respostas deverão ser redigidas exclusivamente com caneta azul ou preta. Respostas a lápis não serão revisadas em hipótese alguma. Respostas com rasuras não serão avaliadas. Não use corretivo. Telefones celulares, agendas, estojos, bolsa e afins não podem ficar sobre as carteiras.</i>		<b>VALOR:</b> 3,0 PONTOS  <b>NOTA:</b>	

### Prova de Fundamentos de Programação

- 1) Indique se as afirmações abaixo são verdadeiras (V) ou falsas (F) em relação às diferenças entre linguagens de programação de alto nível e linguagens de programação de baixo nível. (0.3 pt)
- ( ) Linguagens de alto nível são mais próximas da linguagem humana, facilitando a programação, enquanto linguagens de baixo nível são mais próximas da linguagem de máquina.
  - ( ) Programas escritos em linguagens de baixo nível são mais fáceis de codificar e fazer manutenção do que os programas escritos em linguagens de alto nível.
  - ( ) Linguagens de alto nível são mais abstratas e genéricas, permitindo que os programadores se concentrem mais na lógica do problema em comparação com as linguagens de baixo nível.
  - ( ) O desempenho dos programas escritos em linguagens de baixo nível é geralmente superior ao dos programas escritos em linguagens de alto nível.
- 2) O algoritmo Bubble Sort é um método de ordenação simples que percorre repetidamente a lista, compara cada elemento com o próximo e os troca de posição se estiverem na ordem errada. Esse processo continua até que a lista esteja completamente ordenada. Considere um vetor com os seguintes elementos: [5, 2, 9, 3, 1]. Utilizando o algoritmo Bubble Sort, qual será o estado do vetor após a quarta iteração do algoritmo? (0.3 pt)
- ( ) [1, 2, 3, 5, 9]
  - ( ) [5, 1, 2, 3, 9]
  - ( ) [9, 5, 3, 2, 1]
  - ( ) [1, 3, 2, 5, 9]
  - ( ) [5, 2, 9, 3, 1]

3) Indique se as afirmações abaixo são verdadeiras (V) ou falsas (F) em relação a diferença entre programação orientada a objetos (POO) e programação estruturada. (0.3 pt)

( ) Na programação orientada a objetos, os programas são estruturados em blocos independentes de código chamados "objetos", que contêm dados e comportamentos relacionados, enquanto a programação estruturada organiza o código em sequências lógicas e condicionais.

( ) A programação estruturada utiliza objetos como sua principal unidade de organização do código, enquanto a programação orientada a objetos se baseia em funções e procedimentos para estruturar o programa.

( ) Na programação estruturada, os programas são organizados com base em hierarquias de classes e herança, enquanto a programação orientada a objetos usa estruturas de controle de fluxo para organizar o código de forma eficaz.

( ) Ambas as abordagens, programação orientada a objetos e programação estruturada, são idênticas em sua organização e não diferem significativamente em termos de conceitos e práticas.

4) Indique a alternativa que melhor descreve a importância da indentação no código e da escolha de bons identificadores para as variáveis. (0.3 pt)

( ) A indentação no código e a escolha de bons identificadores para variáveis são meramente preferências estéticas e não afetam a legibilidade ou manutenção do código.

( ) A indentação não é relevante para a compreensão do código, mas a escolha de identificadores descritivos para variáveis é fundamental para tornar o código mais legível e fácil de manter.

( ) A indentação e a escolha de identificadores descritivos são práticas importantes para melhorar a legibilidade e a compreensão do código, facilitando sua manutenção e colaboração entre os desenvolvedores.

( ) A indentação é fundamental para identificar erros de sintaxe no código, enquanto a escolha de identificadores não tem impacto significativo na qualidade do código.

5) Desenvolva um programa que precisa percorrer uma matriz 2x3. 2 representa o número de linhas e 3 representa o número de colunas. Inicialize a matriz para cada elemento ter um dos três valores: 0, 1 ou 2 distribuídos de forma aleatória. Seu algoritmo deve percorrer a matriz e contar quantas células possuem o valor 2. Escreva o algoritmo ou pseudocódigo para essa tarefa. (0.3 pt)

Passo-a-passo:

- Crie uma matriz 2x3, e inicialize distribuindo os valores 0, 1 e 2 aleatoriamente.
- Percorra a matriz e conte quantas células possuem o valor 2.
- Exiba o total de células com o valor 2.

Dica: Em portugol use a função **randi(2)** para gerar valores aleatórios entre 0 e 2. Em python use a função **randint(0,2)**

- 6) Você foi designado para criar um fluxograma que permita ao usuário escolher entre calcular a área de um retângulo ou a área de um círculo. A fórmula para calcular a área de um retângulo é:  $\text{area} = \text{comprimento} \times \text{largura}$ . A fórmula para calcular a área de um círculo é:  $\text{area} = 3.14 \times \text{raio} \times \text{raio}$ . Crie um fluxograma que aceite a escolha do usuário e, com base na escolha feita, peça as entradas apropriadas e gere a área correspondente. (0.3 pt)

Crerérios:

- Solicite ao usuário para escolher entre calcular a área de um retângulo ou a área de um círculo.
- Se o usuário escolher calcular a área do retângulo:
  - Solicite ao usuário para inserir o comprimento do retângulo.
  - Solicite ao usuário para inserir a largura do retângulo.
  - Calcule a área do retângulo usando a fórmula do retângulo.
  - Exiba a área calculada do retângulo.
- Se o usuário escolher calcular a área do círculo:
  - Solicite ao usuário para inserir o raio do círculo.
  - Calcule a área do círculo usando a fórmula do círculo.
  - Exiba a área calculada do círculo.

Você deve descrever as etapas em termos de blocos de ação em um fluxograma, indicando como o programa deve proceder desde a escolha do usuário até a exibição da área calculada correspondente ao retângulo ou ao círculo.

- 7) Escreva um algoritmo que calcule o consumo de energia elétrica de acordo com a categoria de um imóvel (1 - residencial, 2- comercial ou 3 - industrial) e o consumo mensal de energia em kWh (quilowatt-hora). As tarifas de energia são diferentes para cada categoria:

- Residencial: R\$ 0.60 por kWh
- Comercial: R\$ 0.75 por kWh
- Industrial: R\$ 1.20 por kWh

Seu algoritmo deve solicitar a categoria do imóvel e o consumo mensal de energia em kWh como entrada, e calcula o valor a ser pago segundo a fórmula:

$$\text{valor} = \text{consumo} * \text{tarifa}$$

Utilize portugol ou python para elaborar o algoritmo. (0.4 pt)

- 8) Crie um algoritmo que determina a categoria de idade de uma pessoa de acordo com a seguinte tabela:

Idade	Categoria
0 - 12 anos	Criança
13 - 17 anos	Adolescente
18 - 59 anos	Adulto
> 60 anos	Idoso

O algoritmo deve solicitar a idade de uma pessoa como entrada e imprime a categoria correspondente. Escreva o algoritmo em portugol ou em python para resolver esse problema. Caso o programa receba uma idade inválida (número negativo), exiba a mensagem: "Idade inválida!" (0.3 pt)

- 9) Implemente uma função chamada `verificaPalindromo` em portugol ou python que determine se uma sequência de caracteres é palíndroma ou não. A função retorna **verdadeiro/true** se a sequência for palíndroma ou **falso/false** caso contrário. Um palíndromo é uma sequência que permanece a mesma quando lida da esquerda para a direita e vice-versa, ex.: arara e radar. (0.5 pt)

Sugestão de Passo a passo:

1. Receba a sequência de caracteres chamada **palavra** como parâmetro. Suponha que **palavra** contém apenas caracteres minúsculos. (No caso do portugol não é possível passar vetores como parâmetros, então suponha que palavra é uma variável global vetor de 10 posições)
2. Inicialize duas variáveis contadoras para percorrer **palavra**, uma no início da sequência (esquerda) e outra no final (direita).
3. Enquanto o contador da esquerda for menor ou igual o contador da direita:
  - a. Compare os caracteres na posição da variável da esquerda e da direita.
  - b. Se os caracteres forem diferentes, a sequência não é um palíndromo e a repetição deve ser interrompida.

- c. Atualize a posição da variável da esquerda para a direita e a posição da variável da direita para a esquerda.
4. Se a repetição terminar, significa que a sequência é um palíndromo. Retorne verdadeiro/true se a sequência é um palíndromo ou falso/false caso contrário.
- Dica:** para saber o tamanho da lista de caracteres em python use a instrução **len(palavra)**, no caso do portugol crie uma variável **tam** com o tamanho do vetor **palavra**.

10) **(Extra)** Implemente um algoritmo que é responsável por realizar uma técnica básica de compressão. A técnica consiste em comprimir uma sequência de caracteres contando os caracteres repetidos. Por exemplo, a sequência **aabcccccaa** comprimida fica **a2b1c5a3**. Assuma que nas sequência de caracteres processadas pelo algoritmos só há caracteres minúsculos e que os caracteres da sequência estão armazenados em uma lista (python) ou vetor (portugol). Seu algoritmo deve receber uma sequência de caracteres e escrever na tela a sequência comprimida. (0.7 pt)

**ESPAÇO PARA RESOLUÇÃO DAS QUESTÕES**

**ESPAÇO PARA RESOLUÇÃO DAS QUESTÕES**

**ESPAÇO PARA RESOLUÇÃO DAS QUESTÕES**