

Protocolo de Finalização Nº 000083519017

O aluno RAPHAEL FERNANDO REIS DA SILVA com RA **25123537-4** finalizou a atividade **PROVA - FUNDAMENTOS DE PROGRAMAÇÃO COM PYTHON - 41_2025** em **12/01/2025 10:49:36**

PROVA - FUNDAMENTOS DE PROGRAMAÇÃO COM PYTHON - 41_2025**Período:**02/12/2024 08:00 a 18/01/2025 23:59 (Horário de Brasília)**Status:**ABERTO**Nota máxima:**6,00**Nota obtida:****1ª QUESTÃO**

A realização de testes de mesa em algoritmos que contém estruturas de decisão deve considerar o monitoramento de atribuições a variáveis, valores de entrada e saída, constantes e a resolução de expressões. Além disso, é preciso compreender o fluxo do algoritmo analisando o comportamento das condições presentes no código. Considere a realização de um teste de mesa sobre o algoritmo abaixo.

```
Resto=0
```

```
Numero=int(input("Digite um número inteiro:"))
```

```
Resto = Numero % 2;
```

```
if (Resto == 0):
```

```
    print("Condição satisfeita.\n")
```

```
else:
```

```
    print("Condição não Satisfeita.\n")
```

De acordo com os resultados encontrados em seu teste de mesa, avalie as afirmações abaixo.

I - O operador "percento" (%), remete à operação do cálculo do resto da divisão inteira entre dois números.

II - O uso do símbolo % resulta em uma divisão entre os dois números.

III - A mensagem do ELSE será impressa caso o número seja ímpar.

Sendo assim, é correto o que se afirma em:

RESPOSTA: I, II e III.**2ª QUESTÃO**

As variáveis em um algoritmo podem ter escopo global ou local. Uma variável possui escopo local quando é declarada dentro de uma sub-rotina, sendo visível apenas nela, ao passo que uma variável global é declarada antes do algoritmo principal. Assinale a alternativa que corresponde ao que acontece quando são declaradas uma variável global e uma variável local com o mesmo nome.

RESPOSTA: A variável local tem preferência sobre a global dentro da função onde foi declarada.**3ª QUESTÃO**

Para praticar seus novos conhecimentos em linguagem python um aluno fez um pequeno programa. O objetivo deste programa é fazer uma divisão de dois números inteiros e atribuir o resultado em uma variável. Analise o código a seguir:

```
A=int(174)
B=int(2)
C=float()
C = A / B
print (A,"", B, " = ", C)
```

Após analisar o código fonte, assinale a alternativa correspondente do que será impresso na tela.

RESPOSTA: O resultado impresso na tela será $174 / 2 = 87.0$.

4ª QUESTÃO

Sequência de Fibonacci é uma sucessão de números que aparece codificada em muitos fenômenos da natureza. Descrita no final do século 12 pelo matemático italiano Leonardo Fibonacci, ela é infinita e começa com 0 e 1. Os números seguintes são sempre a soma dos dois números anteriores. Portanto: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34?

Conteúdo disponível em <<https://bit.ly/3sGsZhk>> acessado em 20/12/2021

def fib(x):

 if(x == 0 or x == 1):

 return x

 else:

 return fib(x-2) + fib(x-1)

n = 7

resultado = fib(n)

print(resultado)

Realizando o teste de mesa, avalie as alternativas abaixo e assinale a que corresponde à realidade.

RESPOSTA: Quando o algoritmo atingir a execução da linha 15, será impresso o número 13 (treze) na tela.

5ª QUESTÃO

Como aprendido nesta disciplina, há vários caminhos (algoritmos) possíveis para a solução de um problema. Contudo, o desempenho e o consumo de recursos do computador pode variar consideravelmente de um algoritmo para outro. Leia atentamente algumas recomendações para a construção de um algoritmo eficiente e em seguida assinale a alternativa correta.

I. Evitar a declaração de variáveis que não serão utilizadas.

II. Usar estrutura condicional aninhada ou de escolha múltipla no lugar de várias estruturas simples.

III. Não utilize comentários, pois eles aumentam o tamanho do algoritmo e reduzem o desempenho.

IV. Evitar a duplicidade de instruções.

Está correto o que se afirma em:

RESPOSTA:	I, II e IV, apenas.
------------------	---------------------

6ª QUESTÃO

Em termos gerais, a recursão pode ser considerada como um processo de repetição de uma rotina. Portanto, de maneira bem simplista, pode ser definida como uma rotina (procedimento ou função) que chama a si mesma, de forma direta ou indireta.

Bem, se a rotina chama a si mesma inúmeras vezes, é preciso então tomar muito cuidado com o LOOP. Acredito que todos já tenham ouvido falar desse termo, mas esclarecendo para quem não está acostumado: um LOOP ocorre quando uma parte do código fica repetindo eternamente, ocasionando o travamento do sistema todo

Conteúdo disponível em < <https://www.embarcados.com.br/recursividade/> >

Utilize seus conhecimentos adquiridos ao longo da disciplina para interpretar as afirmações abaixo.

I - Uma função recursiva precisa de uma condição de parada ou caso base que, geralmente, é expresso por uma estrutura de decisão.

II - A chamada recursiva de uma função é feita quando a função invoca a si mesma dentro do seu próprio bloco de comandos.

III - A função para calcular o n-ésimo termo da Série de Fibonacci é um exemplo de função inerentemente recursiva.

Sendo assim, é correto o que se afirma em:

RESPOSTA:	I, II e III.
------------------	--------------

7ª QUESTÃO

Uma variável é um recurso utilizado em linguagens de programação (independente de qual seja). De modo geral, uma variável é um espaço reservado na memória do computador que será utilizado no momento que o sistema esteja em execução no dispositivo (seja ele um computador ou não) serve para armazenar dados/informações. Uma variável declarada aceita um determinado tipo de valor a ser armazenado, em python podemos ter quatro tipos principais sendo eles:

RESPOSTA:	Float, Int, Bool e Str.
------------------	-------------------------

8ª QUESTÃO

A passagem de parâmetros serve como um ponto de comunicação entre uma função e a chamada na mesma no meio do código, possibilita a passagem de valores manipulados no código para a função, a função pode ou não se utilizar destes dados passados por parâmetros para executar ações dentro da função. Com base no exposto, vejamos o código a seguir:

```
def ObtemRegiao(uf):  
    if (uf == "PA") or (uf == "AM") or (uf == "AP") or (uf == "RR") or (uf == "AC") or (uf == "RO") or (uf == "TO"):  
        x = "Norte"  
    if (uf == "BA") or (uf == "SE") or (uf == "AL") or (uf == "PE") or (uf == "PB") or (uf == "RN") or (uf == "CE") or (uf == "PI") or (uf == "MA"):  
        x = "Nordeste"  
    if (uf == "GO") or (uf == "MT") or (uf == "MS") or (uf == "DF"):  
        x = "Centro-Oeste"  
    if (uf == "SP") or (uf == "RJ") or (uf == "ES") or (uf == "MG"):  
        x = "Sudeste"  
    if (uf == "PR") or (uf == "SC") or (uf == "RS"):  
        x = "Sul"  
    #return x  
  
uf=str(input("Digite um Estado: "))  
regiao=ObtemRegiao(uf)  
print("Região do Estado: ", regiao)
```

Assinale a alternativa correta sobre o código apresentado é:

RESPOSTA:	Nenhum nome de região será mostrado como resultado na última linha do algoritmo.
------------------	--

9ª QUESTÃO

Como um problema do mundo real, por algumas vezes precisamos tomar decisões que vão nos guiar a caminhos diferentes, isso também acontece em algoritmos de programação. Por exemplo, algumas ações podem ser tomadas com base nas informações que o usuário insere na entrada de dados.

Com base em seus conhecimentos a respeito de estruturas condicionais, analise o algoritmo a seguir.

```
01 - numero = int(input())
```

```
02 - if numero*(-1) < 0:
```

```
03 -     print("Caso 1")
```

```
04 - else:
```

```
05 -     if numero*(-1) > 0:
```

```
06 -         print("Caso 2")
```

```
07 -     else:
```

```
08 -         print("Caso 3")
```

Com base no algoritmo anterior, leia as afirmações a seguir.

I - A linha 03 do algoritmo só será executada caso o usuário insira um valor positivo.

II - A condição da linha 05 só será testada caso o usuário insira um valor não-positivo.

III - A linha 08 será executada quando o número for diferente de zero.

Sendo assim, é correto o que se afirma em:

RESPOSTA: I e II apenas.

10ª QUESTÃO

Observe as afirmativas a seguir e assinale a alternativa correta que contenha apenas elementos válidos dentro da sintaxe de construção de funções.

I - A declaração de parâmetros é a especificação das variáveis de entrada da função em que é preciso especificar o tipo de cada uma das variáveis.

II - O corpo da função é um conjunto de instruções que realizam uma subtarefa, isto é, instruções que realizam o processamento dos dados de entrada e geram a saída de dados dentro de uma função.

III - Toda função que recebe parâmetros não deve retornar valores ao final de sua execução, pois quando se utiliza um dos recursos, automaticamente se abre mão do outro.

IV - Para se implementar o código de uma função com mais de um parâmetro, é necessário que todos tenham os mesmos tipos de dados.

V - O tipo de retorno indica um tipo de valor que a função devolverá ao código chamador ao final de sua execução.

RESPOSTA: I, II e V apenas.