

Fabiana
Rotella
Campanari

 Logout(/impacta/logout/)

(/impacta/schedule/resultcandidatelist/1220beec-
d3d7-11ed-b185-0242ac11002e/)

Registration: 2302122

Candidate: Fabiana Rotella Campanari	Registration: 2302122
Assessment: Prova Oficial Multidisciplinar EAD 100%	Score: 10.00 / 10.00
Date: 02/06/2023 7 p.m.	
Place: Faculdade Impacta / Paraíso / Andar / _Prova Online	
Academic: EAD - ADS 1B - 2° BI / 2023/1 / LP - EAD100 / EAD	

1) Avalie o código da imagem a seguir e marque a alternativa que apresenta as saídas corretas, geradas pelo programa caso o usuário execute-o 3 vezes, com as respectivas entradas:

1)

vertebrado
mamífero

2)

invertebrado
anelídeo

3)

invertebrado
aracnídeo

```

entrada_1 = input()
entrada_2 = input()

if entrada_1 == "vertebrado":
    if entrada_2 == "ave":
        print("é uma ave")
    elif entrada_2 == "mamífero":
        print("é um mamífero")
    else:
        print("vertebrado desconhecido")
elif entrada_1 == "invertebrado":
    if entrada_2 == "inseto":
        print("é um inseto")
    elif entrada_2 == "anelídeo":
        print("é um anelídeo")
    else:
        print("invertebrado desconhecido")
else:
    print("animal desconhecido")

```

- A) é um mamífero é um anelídeo é um aracnídeo
- B) é um mamífero é um inseto é um anelídeo
- C) é uma ave é um inseto é um aracnídeo
- D) é um mamífero é um anelídeo invertebrado desconhecido
- E) é uma ave é um anelídeo invertebrado desconhecido

2) Considerando o programa da figura a seguir, assinale a alternativa correta supondo que o usuário dê como entradas os valores 100 e 200, respectivamente.

```

a = int(input())
b = int(input())
c = 0
while c < a:
    print('bom dia!')
    d = 0
    while d < b:
        print('boa tarde!')
        d += 1
    c += 1
print('boa noite!')

```

- A)** A *string* 'bom dia!' será exibida 100 vezes; a *string* 'boa tarde!' será exibida 100 vezes; a *string* 'boa noite!' será exibida 1 vez.
- B)** A *string* 'bom dia!' será exibida 100 vezes; a *string* 'boa tarde!' será exibida 20000 vezes; a *string* 'boa noite!' será exibida 100 vezes.
- C)** A *string* 'bom dia!' será exibida 100 vezes; a *string* 'boa tarde!' será exibida 20000 vezes; a *string* 'boa noite!' será exibida 1 vez.
- D)** A *string* 'bom dia!' será exibida 100 vezes; a *string* 'boa tarde!' será exibida 200 vezes; a *string* 'boa noite!' será exibida 100 vezes.
- E)** A *string* 'bom dia!' será exibida 100 vezes; a *string* 'boa tarde!' será exibida 200 vezes; a *string* 'boa noite!' será exibida 1 vez.

3) Python tem regras de precedência e associatividade para definir como os operandos são relacionados aos operadores em uma expressão, evidentemente isso influencia no resultado. No início da aprendizagem, é comum inserirmos pares de parênteses desnecessários nas expressões, o que eventualmente compromete a legibilidade do código. Por isso, em geral, recomenda-se utilizar apenas os parênteses essenciais.

São essenciais os pares de parênteses que, se retirados, podem alterar o resultado da expressão. Por exemplo, em $5+(3*2)$, podemos retirar os parênteses sem alterar o resultado, mas na expressão $4*(3+5)$ os parênteses influenciam o resultado da expressão. Entretanto, nas expressões $2+3-4$ e $2+(3-4)$ não há diferença no resultado, mas há alteração na associação dos operandos com os operadores, pois 3 foi associado à subtração por causa dos parênteses.

Você, que estudou o material didático, já conhece bem as regras e sabe como deixar as expressões mais legíveis apenas com os parênteses essenciais e que mantêm a mesma associação entre operandos e operadores. Sabendo disso, veja a expressão a seguir:

$((11\%((5+4)**2))-(12/(2*3)))$

Indique a alternativa que contém uma expressão semelhante à da figura, com o *mesmo resultado e mesma associação entre operandos e operadores*, além de conter apenas os parênteses essenciais.

- A)** $11\%5+4**2-12/2*3$
- B)** $11\%(5+4)**2-(12/(2*3))$
- C)** $11\%(5+4)**2-12/2*3$
- D)** $(11\%(5+4)**2)-(12/(2*3))$
- E)** $11\%(5+4)**2-12/(2*3)$

4) Como visto na disciplina, existem pelo menos dois instrumentos necessários para iniciar a resolução de problemas visando a implementação automatizada da solução em um computador:

(I) Um deles tem o papel de representar os aspectos importantes do problema alvo, descartando o que é irrelevante e objetivando criar um cenário que possa ser entendido pelo programador para ajudar na compreensão do problema e de como resolvê-lo, facilitando, por exemplo, a identificação do estado inicial do problema e qual o estado final desejável;

(II) O outro é uma sequência de instruções logicamente ordenada que, se formulada e executada corretamente, resolve o problema, transformando o estado inicial em um estado final desejável.

Com base no exposto, indique a alternativa que contém os termos mais adequados aos quais (I) e (II) se referem, respectivamente:

- A)** A *representação* e o *algoritmo*.
- B)** O *computador simplificado* e o *processamento*.
- C)** A *entrada* e a *saída*.

- D)** A linguagem de programação e o computador.
 - E)** O algoritmo e a representação.
-

5) Em relação ao escopo de variáveis em Python, conforme visto em nosso material didático, avalie as afirmações a seguir e marque a alternativa com a afirmação FALSA.

- A)** Ao atribuir um valor a uma variável no interior de uma função, será criada pelo Python uma variável local, exceto se for explicitamente indicado um escopo diferente.
 - B)** Todas as chamadas para uma mesma função compartilham o mesmo escopo, então é sempre preciso limpar manualmente os dados da memória em Python para evitar que a função use valores desatualizados para as variáveis locais.
 - C)** Os parâmetros das funções são também variáveis locais da função a qual fazem parte.
 - D)** As variáveis locais têm precedência sobre as variáveis globais, isto é, caso haja na memória uma variável local e uma global com o mesmo nome, por padrão, a preferência da função será pelo uso da variável local.
 - E)** Sobre o escopo local e global, é correto dizer que o escopo global é referente às variáveis criadas na raiz do código (fora de qualquer função), e o escopo local refere-se às variáveis criadas no interior de funções.
-

6) Vimos no material didático que a passagem de argumentos em Python é dita “passagem por objeto” e que isso funciona como uma mistura entre os conceitos de “passagem por referência” e “passagem por valor”. Marque a alternativa correta em relação a como é feita a passagem de argumentos em Python.

- A)** A passagem de argumentos é feita sempre por referência, mas observamos o efeito colateral, típico dessa forma de passagem de argumentos, apenas em objetos que sejam mutáveis. Os objetos imutáveis se comportam como se a passagem tivesse sido por valor.
 - B)** A passagem de argumentos é feita sempre por valor, e uma cópia dos dados é feita sempre.
 - C)** Podemos escolher de qual forma iremos passar os argumentos, por padrão é feita a passagem por valor, e quando queremos fazer a passagem por referência, precedemos a variável por um cifrão.
 - D)** O Python verifica o tipo da variável e então escolhe se irá fazer uma cópia do valor ou passar a referência para a função.
 - E)** Podemos escolher de qual forma iremos passar os argumentos, por padrão é feita a passagem por valor, e quando queremos fazer a passagem por referência, precedemos a variável por um asterisco.
-

7) Considerando um programa que lê dois valores inteiros e exibe o resultado do primeiro número dividido pelo segundo número, escolha a alternativa que o complete:

```
n1 = _____  
n2 = int(input("Informe outro número: "))  
  
print("O resultado da divisão é: {:.2f}".format(n1/n2))
```

- A)** input("Informe um número: ")
 - B)** int(input("Informe um número: "))
 - C)** print(input("Informe um número: "))
 - D)** int("Informe um número: ")
 - E)** float(int("Informe um número: "))
-

8) A respeito das estruturas de seleção composta (if-else), assinale a alternativa correta.

- A)** Tanto o comando if quanto o comando else devem receber uma condição.
- B)** É possível iniciar uma estrutura de seleção composta diretamente pelo comando else, sem usar o comando if antes.

- C)** Apenas o comando if deve receber uma condição, o comando else será executado quando a condição do if for falsa.
- D)** É possível criar uma estrutura de seleção composta com zero condições, deixando tanto o comando if quanto o comando else sem nenhuma condição.
- E)** Apenas o comando if deve receber uma condição, o comando else será executado quando a condição do if for verdadeira.

9) O código a seguir solicita ao usuário que seja digitado números inteiros e os armazene em uma lista de 3 posições. Substituindo todos os valores positivos e iguais a zero por 1 e todos os valores negativos por 0.

```
def troca(lista):  
    for i in range(3):  
        if lista[i] >= 0:  
            lista[i] = 1  
        else:  
            lista[i] = 0  
    return lista  
  
lista = [0]*3  
for i in range(3):  
    lista[i] = int(input('Digite um valor: '))  
print(lista)  
  
print(lista)
```

Selecione a opção que completa o código corretamente.

- A)** troca(lista[])
- B)** lista(troca)
- C)** lista(i)
- D)** troca(range)
- E)** troca(lista)

10) O programa da figura a seguir recebe como entrada um número natural n e deve exibir a sequência de valores naturais de zero até $n-1$ e de $n-1$ até zero. Por exemplo, se $n=3$ o programa deverá exibir 0 1 2 e, na próxima linha, 2 1 0. Assinale a alternativa que contém os valores que substituem adequadamente os símbolos ? (interrogação), e na ordem correta, de modo que o programa funcione conforme a especificação. Observação: Desconsidere as '|' das alternativas, são apenas separadores da sequência de valores.

```
n = int(input('número: '))

inicio = ?
fim     = ?
passo   = ?

for i in range(inicio, fim):
    print(i, end=' ')

print()

for i in range(?, ?, passo):
    print(i, end=' ')
```

- A) 0 | n | -1 | fim + 1 | inicio + 1
- B) 0 | n | -1 | fim | inicio + 1
- C) 0 | n | -1 | fim - 1 | inicio - 1
- D) 0 | n | 1 | fim + 1 | inicio - 1
- E) 0 | n + 1 | -1 | fim - 1 | inicio + 1