AC3 - Linguagem de Programação

Total de pontos 10/10

As questões contidas nessa atividade estão relacionadas aos conteúdos de todas as partes.

O e-mail do participante (**fabiana.campanari@aluno.faculdadeimpacta.com.br**) foi registrado durante o envio deste formulário.

✓ Assinale a alternativa correta em relação a seguinte afirmação: "Em Python usar listas é uma forma de armazenar diversos valores sem ter que criar uma variável particular para cada valor". *1/1

- ○ A frase é verdadeira, porém as listas devem ter tamanho imutável e itens homogêneos, ou seja, todos os itens devem ser do mesmo tipo de dado.
- ◉ A frase é verdadeira e, após criar uma lista com itens, bastará usar o nome dado à lista, isto é, a variável que a referencia, e um par de colchetes em volta de um índice válido para acessar o item daquela posição. ✓
- ○ A frase é verdadeira e, após criar uma lista com itens, bastará usar o nome dado à lista, isto é, a variável que a referencia, e um par de parênteses em volta de um índice válido para acessar o item daquela posição.
- ○ A frase é falsa, pois em Python a única estrutura de dados que permite armazenar diversos itens é a string.
- ○ A frase é falsa, pois em Python não existem listas, apenas arrays, que também são chamados de vetores.

Feedback

Conteúdo abordado no Capítulo 9 da Unidade 3 da apostila.

[CORRETA] A frase é verdadeira e, após criar uma lista com itens, bastará usar o nome dado à lista, isto é, a variável que a referencia, e um par de colchetes em volta de um índice válido para acessar o item daquela posição.

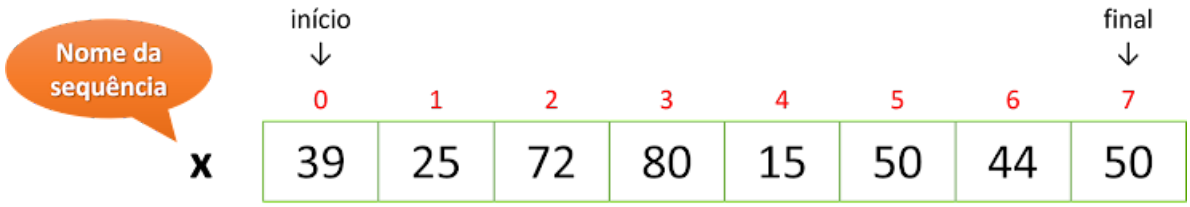
[INCORRETA] A frase é verdadeira e, após criar uma lista com itens, bastará usar o nome dado à lista, isto é, a variável que a referencia, e um par de parênteses em volta de um índice válido para acessar o item daquela posição.
Razão: Um identificador seguido de um par de parênteses é uma chamada de função ou comando da linguagem de programação Python, não uma indexação de sequência.

[INCORRETA] A frase é falsa, pois em Python não existem listas, apenas arrays, que também são chamados de vetores.
Razão: Absurdo! Em Python existem listas.

[INCORRETA] A frase é falsa, pois em Python a única estrutura de dados que permite armazenar diversos itens é a string.
Razão: Em nosso material didático abordamos quatro tipos de estruturas de dados sequenciais e todas elas permitem o armazenamento de diversos itens, caso contrário seria um absurdo por definição, pois são "sequências de itens".

[INCORRETA] A frase é verdadeira, porém as listas devem ter tamanho imutável e itens homogêneos, ou seja, todos os itens devem ser do mesmo tipo de dado.
Razão: Não, já foi explicado em nosso material didático que listas são sequencias potencialmente heterogêneas e com possibilidade de alteração de tamanho.

✓ Analise as afirmações sobre a sequência ilustrada na figura a seguir e selecione a alternativa que está integralmente correta: (I) o tamanho da sequência x é oito, pois o tamanho corresponde à quantidade de itens da sequência. Também é possível garantir que x, por ter tamanho oito, pode ter seu penúltimo item acessado por meio da indexação x[7] ou x[-2]; (II) supondo que x seja uma tupla, é possível modificar o valor de seu primeiro item para 88 com a instrução x[0] = 88, uma vez que as sequências em Python iniciam no índice zero; (III) a sequência x possui dois itens com o mesmo índice, que são os dois com valor 50. *1/1



- Apenas as afirmações II e III estão corretas.
- Todas as afirmações estão corretas.
- (●) Nenhuma das afirmações está correta. ✓
- Apenas as afirmações I e III estão corretas.
- Apenas as afirmações I e II estão corretas.

Feedback

Conteúdo abordado no Capítulo 9 da Unidade 3 da apostila.

Resposta: Nenhuma das afirmações está correta.

[INCORRETA] (I) O tamanho da sequência x é oito, pois o tamanho corresponde à quantidade de itens da sequência. Também é possível garantir que x, por ter tamanho oito, pode ter seu penúltimo item acessado por meio da indexação x[7] ou x[-2].
Razão: O tamanho da sequência x é oito, porém a indexação sempre inicia em zero, portanto os índices estão no intervalo de números naturais [0..7]. Logo, para acessar o penúltimo item, seria possível com as instruções x[6] ou x[-2].

[INCORRETA] (II) Supondo que x seja uma tupla, é possível modificar o valor de seu primeiro item para 88 com a instrução x[0] = 88, uma vez que as sequências em Python iniciam no índice zero.
Razão: Tuplas são sequências imutáveis, portanto não é possível alterar seus itens.

[INCORRETA] (III) A sequência x possui dois itens com o mesmo índice, que são os dois com valor 50.
Razão: Em Python é impossível que dois itens tenham o mesmo índice, pois seria incoerente com o próprio conceito de estrutura de dados sequencial, onde um item pode

ser individualizado e acessado por meio da indexação. A sequência x possui dois itens com VALORES iguais, não com índices iguais.

✓ Considere que o programa ilustrado na figura a seguir deveria solicitar ao usuário um número natural x e uma opção de exibição (P para progressiva e R para regressiva), e então exibir, de acordo com a opção: (I) todos os números naturais no intervalo fechado decrescente [x..1] ou; (II) todos os números naturais no intervalo fechado crescente de [1..x]. Porém, a codificação está errada. Selecione a alternativa que contém uma afirmação correta sobre o(s) erro(s) do código-fonte. *1/1

```
def contagem_progressiva(n):
    for i in range(1, n, 1):
        print(i)

def contagem_regressiva(n):
    for i in range(n, 1, -1):
        print(i)

x = int(input('x? '))
opcao = input('Opção? ')
if opcao in 'Pp':
    contagem_progressiva(x)
else:
    contagem_regressiva(x)
```

- A função contagem_progressiva está errada, pois o intervalo usado no laço for deveria ser range(0, n+1).
- A função contagem_progressiva está errada, pois o intervalo usado no laço for deveria ser range(0, n).
- (●) A função contagem_regressiva está errada, pois o intervalo usado no laço for deveria ser range(n, 0, -1). ✓
- A função contagem_regressiva está errada, pois o intervalo usado no laço for deveria ser range(n, -1, -1).
- A função contagem_progressiva está errada, pois o intervalo usado no laço for deveria ser range(0, n+1, 1).

Feedback

Conteúdo abordado no Capítulo 9 da Unidade 3 e no Capítulo 10 da Unidade 4 da apostila.

Para criar um intervalo decrescente em Python que represente corretamente o intervalo matemático de números inteiros [x..1] basta usar a instrução range(x, 0, -1).

Para criar um intervalo crescente em Python que represente corretamente o intervalo matemático de números inteiros [1..x] basta usar a instrução range(1, x+1).

✓ Em relação à codificação em Python a seguir, assinale a alternativa que indica corretamente qual o tipo de cópia feita em cada variável em relação à lista original L1. *1/1

```
import copy

L1 = [10, 20, 30, [40, 50], 60]
L2 = L1
L3 = copy.copy(L1)
L4 = L1[:]
L5 = L1.copy()
L6 = list(L1)
L7 = copy.deepcopy(L1)
```

- ○ L2 = cópia da referência à lista original; L3 = cópia rasa da lista original; L4 = cópia rasa da lista original; L5 = cópia rasa da lista original; L6 = cópia da referência à lista original; L7 = cópia profunda da lista original.
- ◉ L2 = cópia da referência à lista original; L3 = cópia rasa da lista original; L4 = cópia rasa da lista original; L5 = cópia rasa da lista original; L6 = cópia rasa da lista original; L7 = cópia profunda da lista original. ✓
- ○ L2 = cópia da referência à lista original; L3 = cópia da referência à lista original; L4 = cópia rasa da lista original; L5 = cópia rasa da lista original; L6 = cópia rasa da lista original; L7 = cópia profunda da lista original.
- ○ L2 = cópia da referência à lista original; L3 = cópia rasa da lista original; L4 = cópia da referência à lista original; L5 = cópia rasa da lista original; L6 = cópia rasa da lista original; L7 = cópia profunda da lista original.
- ○ L2 = cópia rasa da lista original; L3 = cópia rasa da lista original; L4 = cópia rasa da lista original; L5 = cópia rasa da lista original; L6 = cópia rasa da lista original; L7 = cópia profunda da lista original.

Feedback

Conteúdo abordado no Capítulo 12 da Unidade 4 da apostila.

Também poderia ter sua execução simulada por meio do Python Tutor onde evidenciaria as consequências de cada instrução.

✓ Considere que o núcleo de uma matriz é a parte da matriz que está entre a primeira e a última linha e entre a primeira e a última coluna. Veja na figura a seguir um exemplo de matriz e seu núcleo destacado em amarelo. Selecione a alternativa que contém uma função que devolve como resposta a soma dos itens do núcleo de uma matriz numérica M de L linhas por C colunas, caso exista mais de uma função correta dentre as opções, selecione a alternativa que indica exatamente quantas são. *1/1

Matriz numérica 5 x 7 (5 linhas por 7 colunas)

68	86	50	42	84	98	75
15	83	10	16	14	99	58
30	68	78	44	70	25	31
69	53	17	87	55	83	15
20	70	78	61	65	16	26

A soma do núcleo da matriz é 802

```
def soma_nucleo(M, L, C):
    soma = 0
    for i in range(1, L):
        for j in range(1, C):
            soma += M[i][j]
    return soma
```

Função 3, somente esta função está correta.

```
def soma_nucleo(M, L, C):
    soma = 0
    for i in range(1, L-1):
        for j in range(1, C-1):
            soma += M[i][j]
    return soma
```

Função 2, somente esta função está correta. ✓

As três funções estão corretas.

Há duas funções corretas dentre as alternativas.

```
def soma_nucleo(M, L, C):
    soma = 0
    for i in range(2, L-1):
        for j in range(2, C-1):
            soma += M[i][j]
    return soma
```

- ○ Função 1, somente esta função está correta.

Feedback

Conteúdo abordado no Capítulo 12 da Unidade 4 da apostila.

Basta analisar as codificações de cada alternativa e/ou verificar a resposta de cada função utilizando a própria linguagem Python.

✓ Uma empresa estava enfrentando problemas em seu setor financeiro, atrasando pagamentos e causando insatisfação em seus funcionários. Para evitar um grande desastre, o diretor da empresa resolveu reformular a área problemática e, para recompensar seus funcionários, estabeleceu que todos ganhariam uma aumento de 10% no salário. Para ser mais generoso, o diretor confirmou que os funcionários que obtiveram uma avaliação positiva dos clientes da empresa também receberão um gratificação de 5% do salário (já atualizado) por cada ano que tenham completado na empresa. Como o diretor estava muito ocupado, pediu para um colega desenvolver um programa em Python que: (I) criasse uma tabela com os dados dos funcionários (nome, salário antes do aumento, quantidade de anos na empresa e um valor indicando se o funcionário obteve uma avaliação positiva dos clientes); (II) atualizasse na tabela o salário dos funcionários; (III) adicionasse um campo na tabela com as gratificações concedidas, inclusive quando for zero e; (IV) exibisse ao final o investimento que a empresa fez com o aumento dos salários e gratificações. Entretanto, ao executar o programa, foi constatado que o programa não funcionava corretamente. Assinale a alternativa que contém uma afirmação FALSA sobre os erros desse programa. *1/1

```
# Campos da tabela (nesta ordem):
# Nome | Salário | Anos na empresa | Avaliação positiva
funcionarios = [['Ana',  3500.00,  5, True ],
                ['Bia',  2450.00,  3, False],
                ['Clô',  6700.50, 10, False],
                ['Duda', 1212.00,  1, True ],
                ['Elen', 5000.00,  2, True ]]

investimento = 0

for funcionario in funcionarios:
    aumento = 0.10 * funcionario[2]
    funcionario[2] += aumento
    investimento += aumento
    if funcionario[3]:
        funcionario.append(0.50 * funcionario[2] * funcionario[1])
    investimento += funcionario[5]

print(f'Investimento: R$ {investimento:.2f}')
```

- ○ O programa deveria acrescentar um campo na tabela com o valor da gratificação para cada funcionário, porém só adiciona a coluna naqueles que foram avaliados positivamente pelos clientes, isto é, aqueles com True no respectivo campo.
- ○ Nesta tabela não existirá um campo com índice 5, mesmo após adicionar o campo com o valor da gratificação.

- (●) Existem dados nos registros da tabela que estão armazenados com o tipo incorreto, por exemplo o campo que está no índice 1. ✓
- ( ) O aumento deveria ser calculado com base no valor do salário, porém a instrução que calcula o aumento considerou o campo errado do registro.
- ( ) O valor da gratificação foi calculado incorretamente.

Feedback

Conteúdo abordado no Capítulo 12 da Unidade 4 da apostila.

A única alternativa incorreta é "Existem dados nos registros da tabela que estão armazenados com o tipo incorreto, por exemplo o campo que está no índice 1", pois o campo que está no índice 1 dos registros é o que corresponde ao salário e deve ser float, exatamente o tipo de dados que está armazenado.

✓ Leia e analise a seguinte afirmação: "Em Python uma lista é uma estrutura que armazena os dados em sequência, podendo os itens serem acessados por meio de seus índices que são naturalmente ordenados de zero até o tamanho da lista-1. Vale ressaltar que esses dados podem ser de tipos diferentes, isto é, as listas são estruturadas de dados potencialmente heterogêneas.". Agora marque a alternativa correta em relação à esta afirmação. *1/1

- ○ A frase é falsa, pois em Python as listas devem ser homogêneas, ou seja, os itens devem ser de um mesmo tipo.
- ◉ A frase está correta, pois listas são sequencias e, por definição, tem seus itens associados com índices numéricos que estão em sequência, independentemente da ordem dos valores dos itens (itens crescentes, decrescentes ou sem ordem definida). ✓
- ○ A frase é falsa, pois em Python as listas têm índices arbitrários, ou seja, o primeiro item pode ter um índice superior ao do segundo que, por sua vez, pode ter um índice inferior ao terceiro e assim por diante.
- ○ A frase é verdadeira, porém em Python só pode-se criar listas que tenham os valores dos itens em ordem crescente ou decrescente, jamais desordenados.
- ○ Não é possível afirmar que a frase é verdadeira ou falsa, pois em Python as listas são estruturas de dados recentes e que não foram bem definidas pelo criador da linguagem de programação.

Feedback

Conteúdo abordado no Capítulo 9 da Unidade 3 da apostila.

Basta verificar o material didático, a afirmação e as alternativas fazem parte da definição de "Sequencias" e de "Listas".

✓ Marque a alternativa que apresenta, respectivamente, uma forma correta de se adicionar, acessar, modificar e remover um valor v em uma lista, através do seu índice i. *1/1

- ○ lista.append(i, v) ; lista[i] ; lista[i] = v ; lista.pop(v)
- ◉ lista.insert(i, v) ; lista[i] ; lista[i] = v ; lista.pop(i) ✓
- ○ lista.append(i) ; lista(i) ; lista(i) = v ; lista.remove(i)
- ○ lista.insert(i) ; lista[i] ; lista[i] = v ; lista.remove(i)
- ○ lista.insert(v) ; lista[i] ; lista[i] = v ; lista.pop(i)

Feedback

Conteúdo abordado no Capítulo 10 da Unidade 4 da apostila.

lista.insert(i, v) --> insere um valor v no índice i;

lista[i] --> acessa o valor que está no índice i;

lista[i] = v --> atribuí o valor v ao item no índice i da lista;

lista.pop(i) --> remove da lista o valor no índice i e retorna o valor removido.

✓ Em Python, é possível buscar um valor em uma sequência usando o método s.index(v), que retorna o índice da primeira ocorrência do valor v na sequência s. No entanto, não há nenhum método que permita a busca de todos os índices quando há mais de uma ocorrência do mesmo valor. O código da imagem a seguir implementa a função encontra_indices() para fazer exatamente isso, retornar uma sequência contendo todos os índices nos quais o valor v é encontrado na sequência s. Avalie-o e marque a alternativa CORRETA. *1/1

```
1 def encontra_indices(v, s):
2     indices = []
3     for i, item in enumerate(s):
4         if item == v:
5             indices.append(i)
6
7     return indices
8
```

- ○ Há um erro na linha 5, pois listas são imutáveis e não possuem o método append().
- ○ Há um erro na linha 7, pois quando o valor buscado não existe na sequência s a variável índices não é criada.
- ○ A função possui um erro de lógica, e não retorna os índices corretos, referentes ao valor v buscado.
- ◉ A função está correta e irá retornar todos os índices de um valor v buscado em uma sequência s. ✓
- ○ Há um erro na linha 3, pois não é possível fazer o desempacotamento na mesma linha de uma instrução de laço for

Feedback

Conteúdo abordado no Capítulo 9 da Unidade 3 e nos capítulos 10 e 11 da Unidade 4 da apostila.

A função está correta e irá retornar todos os índices da sequência s cujos itens são iguais à v.

A função encontra_indices percorre a enumeração da sequência s, obtendo o índice e o valor de cada item, verifica se o item é igual ao valor e, em caso afirmativo, adiciona o valor desse índice ao final de uma lista local, que é retornada como resposta ao final da execução da função.

Observação: Veja a codificação 11.15 do texto base 11 para entender melhor como funciona o enumerate e o desempacotamento no próprio for.

✓ Leia as afirmações a seguir a respeito do algoritmo de ordenação por flutuação "Bubble Sort", considerando a implementação que foi dada em nosso material didático, e marque a alternativa CORRETA: (I) a ordenação por flutuação é o algoritmo de ordenação mais eficiente conhecido atualmente e é empregado na ordenação usada pelo Python na função sorted e no método sort; (II) ao utilizá-lo é garantida a estabilidade da sequência ordenada, isto é, itens com o mesmo valor são mantidos na mesma ordem relativa entre si; (III) O algoritmo Bubble Sort não exige a criação de outra lista para ordenar a lista original, isto é, é possível realizar uma ordenação in-place sem criar uma lista auxiliar. *1/1

- (●) São verdadeiras apenas as afirmações II e III. ✓
- ( ) São verdadeiras apenas as afirmações I e II.
- ( ) Todas as afirmações são verdadeiras.
- ( ) Nenhuma das afirmações é verdadeira.
- ( ) São verdadeiras apenas as afirmações I e III.

Feedback

Conteúdo abordado no Capítulo 11 da Unidade 4 da apostila.

(I) Falsa: O algoritmo de ordenação por flutuação (bubble sort) não tem um bom desempenho quando utilizado com sequências com muitos itens, pois é pouco eficiente. O algoritmo implementado nas funções do Python é conhecido como Tim Sort, em homenagem a Tim Peters, que o desenvolveu.

(II) Verdadeira: Na implementação apresentada no material didático o Bubble Sort é estável, pois só trocará um par de itens de lugar quando o primeiro item do par for maior do que o segundo, portanto itens de mesmo valor manterão suas posições relativas.

(III) Verdadeira: Basta verificar o algoritmo do Bubble Sort apresentado no material didático e comprovar que não são necessárias outras listas além da original para executar a ordenação.

Google Formulários