

Estudante: _____

1. Considere as seguintes afirmações. Classifique-as em verdadeiras (V), falsas (F) ou não sei (?). Caso não seja classificada, assume-se a classificação não sei (?). Para cada afirmação classificada corretamente como V ou F, a nota será acrescida em 0,6. Para cada afirmação incorretamente classificada, a nota reduzirá em 0,6. Para cada afirmação classificada como não sei, a nota será reduzida em 0,3. **(Valor máximo: 3,0; Valor mínimo: 0,0).**

(F) a) O algoritmo é o conjunto de instruções e regras que definem como resolver um problema ou executar uma tarefa, enquanto o programa é a implementação desse algoritmo em uma linguagem de programação específica. **Para um problema, existe no máximo um algoritmo que o resolve.** Para cada algoritmo, podem existir vários programas que o implementam.

(V) b) Em Python, uma expressão pode combinar operadores aritméticos, relacionais, identidade, lógicos (bit a bit) e booleanos. Operações agrupadas com parênteses possuem a maior precedência para sua resolução. Além disso, temos as seguintes precedências das operações. Com relação aos operadores aritméticos, a precedência é, em ordem de maior prioridade: exponenciação; multiplicação, divisão e módulo; soma e subtração. Para os operadores lógicos (bit a bit), a ordem é: negação; deslocamento para esquerda, deslocamento para direita; e; ou exclusivo; ou. Para os operadores booleanos, a ordem é: not; and; or. Tanto para os operadores relacionais e de identidade, seus operadores possuem a mesma ordem de precedência. Além disso, é necessário considerar que, em uma mesma expressão, é possível utilizar todos os tipos de operadores disponíveis em Python. Isso implica que também existe uma ordem de precedência entre os tipos de operadores. Assim, agrupar as operações com parênteses permite deixar explícita a ordem com que as operações da expressão devem ser resolvidas, sem depender da avaliação da interação entre as diversas operações e suas ordens de precedência.

(V) c) Uma variável em Python pode ser criada, consultada e excluída. A criação é realizada com o comando de atribuição, utilizando a operação "=". A exclusão pode ser feita com o uso da função "del" ou automaticamente, conforme o encerramento do escopo em que a variável foi criada. A consulta à variável é feita sempre que a variável é utilizada em uma expressão. Também existe a possibilidade da combinação da criação e consulta da variável, com os comandos de atribuição combinados com operações (por exemplo, com os operadores "+=" e "*=").

(V) d) Uma função em Python é criada com o comando "def", seguido do nome da função e da definição de seus parâmetros em parênteses. Este nome e os parâmetros constituem a assinatura da função. Após essa assinatura, na linha imediatamente abaixo, devemos definir um bloco de comandos, contendo ao menos um comando. De modo geral, espera-se que uma função implemente uma responsabilidade, ou seja, uma funcionalidade ou propriedade útil para o programa.

(V) e) Um comando de seleção permite alterar o fluxo de execução de um programa a partir da avaliação de uma expressão com resultado Booleano. Em Python, temos dois tipos de comandos de seleção: "if" e "match". Com o "if", podemos ter também o uso do "elif" e do "else". Com o "match", podemos ter vários "case". Uma outra forma de alterar o fluxo de execução de um programa em Python é com uma chamada de função, mas ela não é considerada um comando de seleção, porque a mudança do fluxo de execução não envolve uma avaliação do estado do programa, ou seja, uma seleção entre dois ou mais caminhos possíveis (tal como temos com os comandos "if" e "match").

2. Considere o seguinte programa e responda os itens desta questão. (Valor: 5,0)

Linha	Programa
1	print("Calculadora de desconto. Escolha a forma de pagamento:")
2	print("1 - PIX (15% de desconto)")
3	print("2 - Débito (5% de desconto)")
4	print("3 - Crédito (sem desconto)")
5	valor_compra = float(input("Digite o valor da compra: R\$ "))
6	forma_pagamento = int(input("Digite a forma de pagamento (1-3): "))
7	if valor_compra < 0:
8	print("Erro: O valor da compra deve ser positivo!")
9	elif not (forma_pagamento >= 1 and forma_pagamento <= 3):
10	print("Erro: Forma de pagamento inválida!")
11	else:
12	if forma_pagamento == 1:
13	desconto = 0.15
14	elif forma_pagamento == 2:
15	desconto = 0.05
16	else:
17	desconto = 0.0
18	valor_desconto = valor_compra * desconto
19	valor_final = valor_compra - valor_desconto
20	print("Resumo da Compra:")
21	print("Valor inicial: R\$ %0.2f" % valor_compra)
22	print("Desconto aplicado: R\$ %0.2f" % valor_desconto)
23	print("Valor final: R\$ %0.2f" % valor_final)
24	print("Obrigado! Volte sempre!")

a) Descreva, em uma frase, o que faz este programa. (Valor: 0,75)

O programa calcula o valor de produto após desconto conforme a forma de pagamento.

b) Considere que o usuário informe um produto de valor R\$ 200,00 e pagamento por cartão de débito. Quais linhas seriam executadas por esse programa e quais seriam os resultados mostrados em tela? (Valor: 0,5)

As linhas executadas seriam: 1-6, 7, 9, 11, 12, 14, 15, 18-23, 24.

O resultado completo mostrado na tela seria:

Calculadora de desconto. Escolha a forma de pagamento:

1 - PIX (15% de desconto)

2 - Débito (5% de desconto)

3 - Crédito (sem desconto)

Digite o valor da compra: R\$

Digite a forma de pagamento (1-3):

Resumo da Compra:

Valor inicial: R\$ 200.00

Desconto aplicado: R\$ 10.00

Valor final: R\$ 190.00

Obrigado! Volte sempre!

O resultado simplificado mostrado na tela seria:

Valor inicial: R\$ 200.00

Desconto aplicado: R\$ 10.00

Valor final: R\$ 190.00

c) Considere que o usuário informe um produto de valor R\$ 200,00 e a opção 0 para forma de pagamento. Quais linhas seriam executadas por esse programa e quais seriam os resultados mostrados em tela? **(Valor: 0,5)**

As linhas executadas seriam: 1-6, 7, 9, 10, 24.

O resultado completo mostrado na tela seria:

Calculadora de desconto. Escolha a forma de pagamento:

1 - PIX (15% de desconto)

2 - Débito (5% de desconto)

3 - Crédito (sem desconto)

Digite o valor da compra: R\$ 200

Digite a forma de pagamento (1-3): 0

Erro: Forma de pagamento inválida!

Obrigado! Volte sempre!

O resultado simplificado mostrado na tela seria:

(nada, apenas linhas com valores fixos seriam mostradas)

d) Considere que o usuário informe um produto de valor R\$ -50,00 e a opção 3 para forma de pagamento. Quais linhas seriam executadas por esse programa e quais seriam os resultados mostrados em tela? **(Valor: 0,5)**

As linhas executadas seriam: 1-6, 7, 8, 24.

O resultado completo mostrado na tela seria:

Calculadora de desconto. Escolha a forma de pagamento:

1 - PIX (15% de desconto)

2 - Débito (5% de desconto)

3 - Crédito (sem desconto)

Digite o valor da compra: R\$ -50

Digite a forma de pagamento (1-3): 3

Erro: O valor da compra deve ser positivo!

Obrigado! Volte sempre!

O resultado simplificado mostrado na tela seria:

(nada, apenas linhas com valores fixos seriam mostradas)

e) Este programa contém ao menos três partes, com distintas responsabilidades. Quais são elas? Identifique-as, especificando a faixa de linhas (linha inicial e linha final) e qual é a responsabilidade de cada parte. **(Valor: 1,75)**

O programa poderia ser organizado em três partes dessa forma:

- Entrada de dados: linhas 1 à 6
- Tratamento de erro quanto aos dados de entrada: linhas 7 à 10
- Cálculo do desconto: 12 à 23

Também seria possível separar o cálculo do desconto da apresentação dos resultados (valores original e com desconto, além do desconto propriamente dito). Nesse caso, o cálculo seria das linhas 12 à 19 e a apresentação dos resultados das linhas 20 à 24. Finalmente, ainda quanto ao cálculo do desconto, o desconto em si poderia ser uma responsabilidade distinta do cálculo do valor atualizado do produto. Nesse caso, o cálculo do desconto corresponderia às linhas 12 à 17 e o cálculo do valor atualizado às linhas 18 e 19.

f) Caso você fosse substituir uma das partes do programa por uma função, como seria definida essa função? Descreva as alterações necessárias, com especial atenção para a assinatura da nova função, quais seriam seus parâmetros e seu valor de retorno e em que linha ela seria chamada (e como seria chamada).

(Valor: 1,00)

O cálculo do desconto (linhas 12 à 17) poderia ser uma função. Nesse caso, seria necessário um parâmetro: a forma de pagamento. O corpo da função teria o conteúdo das linhas 12 à 17, retornando-se o valor da variável desconto. A assinatura da função poderia ser **calcular_desconto(forma_pagamento)**. A chamada à essa função substituiria as linhas 12 à 17 da seguinte forma: **desconto = calcular_desconto(forma_pagamento)**.

3. Considere o seguinte programa. Ele calcula a distância euclidiana entre dois pontos. Responda às seguintes questões: **(Valor: 3,0)**

```
1 def distância_euclidiana(x1, x2, y1, y2):
2     delta_x = x1 - x2
3     delta_y = y1 - y2
4     distância = (delta_x**2 + delta_y**2) ** 0.5
5     return distância
6
7 a_x = int(input("Coordenada x do ponto A: "))
8 a_y = int(input("Coordenada y do ponto A: "))
9 b_x = int(input("Coordenada x do ponto B: "))
10 b_y = int(input("Coordenada y do ponto B: "))
11 distancia_a_b = distância_euclidiana(a_x, b_x, a_y, b_y)
12 print("A distância entre A e B é de %.2f" % distancia_a_b)
```

a) Quais são as variáveis globais deste programa? **(Valor: 1,0)**

As variáveis globais são: `a_x`, `a_y`, `b_x`, `b_y`, `distância_a_b`.

b) Quais são as variáveis locais deste programa? **(Valor: 1,0)**

As variáveis locais são aquelas da função `distância_euclidiana`: `x1`, `x2`, `y1`, `y2`, `delta_x`, `delta_y`, `distância`.

c) Considere que o usuário informou os seguintes valores de entrada para o programa: 3, 3, 7, 3. Quais são os valores das variáveis "delta_x" e "delta_y" durante a execução na linha 4? E na linha 12? Explique. **(Valor: 1,0)**

Na linha 4, o valor de `delta_x` será -4 e o valor de `delta_y` será 0. Na linha 12, essas variáveis não estão disponíveis, logo ocorrerá um erro de execução do programa.