

Estudante: _____

1. Associe cada conceito abaixo com uma afirmação. (Valor: 1,0)

- a. Algoritmo
- b. Programa de computador
- c. Implementação
- d. Depuração

(d) Execução controlada de um programa, geralmente oferecida em ambientes de programação, que permite uma acurada investigação sobre falhas e erros de programas.

(b) Representação formal de um algoritmo em uma linguagem de programação, com a definição de comandos e estruturas de dados para processar dados obtidos de dispositivos de entrada ou em memória, com eventual apresentação dos resultados da computação para dispositivos de saída ou armazenamento de resultados em memória.

(a) Descrição de uma sequência de comandos que devem ser seguidos para a realização de uma tarefa.

(c) Escrita de programa de computador, contemplado um ou mais algoritmos, empregando uma linguagem de programação de uso geral e um ambiente apropriado de desenvolvimento.

2. Considere o seguinte programa e responda os itens desta questão: (Valor: 3,0)

Linha	Programa
1	ra = input("Informe o último dígito de seu RA")
2	a = float(ra)
3	b = (3 + ((17 // 3) * 4**2) % 2)
4	c = int(-b / 2) + round(a)
5	c /= b

a) Quais são os valores das variáveis 'a', 'b' e 'c' após a execução da linha 4?

Para "ra" com o valor de "9", teremos os seguintes valores:

a = 9.0

b = 3

c = 8

b) O que faz o comando da linha 5? Explique e mostre o resultado e o tipo de dado deste resultado.

O comando da linha 5 faz a divisão do valor atual da variável c pelo valor da variável b. O resultado desse cálculo é utilizado para atualizar o valor da variável c. Assim, após a execução dese comando, o valor da variável c será 2.666665, do tipo float.

c) Quais são as variáveis definidas e seu tipo de dado durante a execução do programa? Informe isso conforme a execução linha a linha do programa.

Linha	ra	a	b	c
1	"9" (string)			
2	"9" (string)	9.0 (float)		
3	"9" (string)	9.0 (float)	3 (int)	
4	"9" (string)	9.0 (float)	3 (int)	8 (int)
5	"9" (string)	9.0 (float)	3 (int)	2.66665 (float)

3. A sequência de Fibonacci consiste em uma sequência infinita de números inteiros, cujos dois primeiros números são 0 e 1 e os demais números são a soma dos dois antecessores. Assim, o início desta sequência possui estes números: 0, 1, 1, 2, 3, ... Com a fórmula de Binet, é possível calcular o n-ésimo elemento dessa sequência sem calcular os dois termos antecessores. O código abaixo contém um programa que define e usa essa função, mas, infelizmente, o código está fora de ordem (embora corretamente indentado). Corrija e calcule os sétimo e décimo terceiro elementos da sequência de Fibonacci. **(Valor: 3,0)**

Dica: Como o código está correto, embora fora de ordem, todas as variáveis estão corretamente definidas, bastando definir cada variável na ordem correta para suprir a dependência dos comandos seguintes.

```

1  print("The %d-th Fibonacci number is %d" % (number, result))
2      sqrt_5 = math.sqrt(5)
3      fibonacci_number = int((phi ** n - psi ** n) / sqrt_5)
4  import math
5      phi = (1 + sqrt_5) / 2
6      return fibonacci_number
7  def nth_fibonacci_number(n):
8  number = int(input("Informe o número da sequência de Fibonacci a ser calculado:"))
9      psi = (1 - sqrt_5) / 2
10 result = nth_fibonacci_number(number)

```

```

4  import math
7  def nth_fibonacci_number(n):
2      sqrt_5 = math.sqrt(5)
5      phi = (1 + sqrt_5) / 2
9      psi = (1 - sqrt_5) / 2
3      fibonacci_number = int((phi ** n - psi ** n) / sqrt_5)
6      return fibonacci_number
8  number = int(input("Informe o número da sequência de Fibonacci a ser calculado:"))
10 result = nth_fibonacci_number(number)
1  print("O número de Fibonacci %d é %d" % (number, result))

```

Observe que podem existir variações dessa solução que ainda estão corretas. Por exemplo, as linhas 5 e 9 podem ser invertidas; a linha 8 pode ser colocada antes da linha 7.

Para o o sétimo elemento da sequência de Fibonacci, temos o seguinte resultado:

```
Informe o número da sequência de Fibonacci a ser calculado:7
O número de Fibonacci 7 é 13
```

Para o o décimo terceiro elemento da sequência de Fibonacci, temos o seguinte resultado:

```
Informe o número da sequência de Fibonacci a ser calculado: 13
O número de Fibonacci 10 é 233
```

4. A distância entre dois pontos pode ser calculada considerando o teorema de Pitágoras, no que denominamos como distância euclidiana. Cada ponto possui uma posição (x, y) no plano e a distância é a medida do segmento de reta que liga os dois pontos. Para o ponto inicial A teríamos então a posição (x_A, y_A) e para o ponto final B teríamos (x_B, y_B). A distância d seria calculada pela raiz quadrada da soma dos quadrados das diferenças entre x_B e x_A e entre y_B e y_A, ou seja, $d = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$. No quadro a seguir é apresentado o código de um programa que deveria calcular e mostrar a distância entre os pontos A e B, informados pelo usuário, representados pelas variáveis a_x, a_y, b_x e b_y. Esse programa define e faz uso da função `distância_euclidiana` para calcular a distância. No entanto, o código tem alguns erros. Identifique e corrija os erros. (Valor: 3,0)

```
1 import everything from math
2
3 a_x = input(int("Coordenada x do ponto A: "))
4 a_y = input(int("Coordenada y do ponto A: "))
5 a_x = input(int("Coordenada x do ponto B: "))
6 a_y = input(int("Coordenada y do ponto B: "))
7 distancia_a_b = distância_euclidiana()
8 print("A distância entre A e B é de .2f" % distancia_a_b)
9
10 def distância_euclidiana(1a, 2a, 1b, 2b):
11     delta_x = 1b - 1a
12     delta_y = 2b - 2a
13     distância = math.sqrt(delta_x**2 + delta_y**2)
```

Erros do programa:

- Linha 1: A importação do módulo *math* foi realizada incorretamente. Para este programa, basta importar o módulo, sem especificar elementos específicos dele: `import math`
- Linhas 3, 4, 5 e 6: Todas essas linhas possuem o mesmo tipo de erro: foi invertida a ordem de chamada das funções. O correto seria chamar primeiro a função `input` e depois a `int`: `int(input(""))`
- Linha 5: A variável que deveria ser definida é `b_x` ao invés de `a_x`.
- Linha 6: A variável que deveria ser definida é `b_y` ao invés de `a_y`.
- Linha 7: A chamada à função `distância_euclidiana` não apresenta os quatro argumentos especificados na assinatura da função: `distancia_a_b = distância_euclidiana(a_x, a_y, b_x, b_y)`
- Linha 8: Na parametrização da string formatada, faltou adicionar o `%` antes do `.2f`: `print("A distância entre A e B é de %.2f" % distancia_a_b)`
- Linha 10: O nome dos parâmetros começa com um número, mas um nome não pode começar com um número em Python. Uma solução seria inverter, ficando: `def distância_euclidiana(a1, a2, b1, b2)`. Observe que as linhas 11 e 12 devem ser alteradas para refletir os novos nomes dos parâmetros.
- Linhas 10 à 13: A função está definida no local incorreto. Ela precisa ser definida antes de sua chamada na linha 8 ou, melhor ainda, antes de qualquer outro comando diferente de importação do corpo principal do programa. Ou seja, o ideal seria definí-la na linha 2.
- Linha 13: O cálculo do quadrado da diferença dos valores y dos pontos A e B está incorreto (foi calculado a multiplicação por dois ao invés da potência de dois): `distância = math.sqrt(delta_x**2 + delta_y**2)`
- Linha 13-14: A função não está retornando um valor, mas o programa espera que seja retornada a distância euclidiana. Assim, é necessário acrescentar uma linha à função: `return distância`.

Programa correto:

```
1  import math
2
10 def distância_euclidiana(a1, a2, b1, b2):
11     delta_x = b1 - a1
12     delta_y = b2 - a2
13     distância = math.sqrt(delta_x**2 + delta_y**2)
    return distância
9
3  a_x = int(input("Coordenada x do ponto A: "))
4  a_y = int(input("Coordenada y do ponto A: "))
5  b_x = int(input("Coordenada x do ponto B: "))
6  b_y = int(input("Coordenada y do ponto B: "))
7  distancia_a_b = distância_euclidiana(a_x, a_y, b_x, b_y)
8  print("A distância entre A e B é de %.2f" % distancia_a_b)
```