

### Questão 1

Em um programa de computador, informações como números e textos precisam ser armazenadas de alguma forma. Para isso, utilizamos **variáveis**, que funcionam como "caixinhas" onde guardamos dados. Cada variável pode armazenar um tipo diferente de informação, e em **Python** os principais tipos de dados são:

- **int (inteiro)** → Armazena números inteiros, como 5, 10, -3.
- **float (decimal)** → Armazena números com parte decimal, como 2.5, 3.14, -0.1.
- **str (texto ou string)** → Armazena palavras e frases, como "Python", "123", "Olá, mundo!". Note que as strings estão entre aspas.
- **bool (booleano)** → Armazena apenas dois valores: True (verdadeiro) ou False (falso).

A seguir são definidas em código 5 variáveis que representam alunos de uma turma.

```
1 S1 = "Carlos"
2 S2 = "Ana"
3 S3 = "Bruno"
4 S4 = "Eduardo"
5 S5 = "Daniel"
6
7 print(S3)
```

→ Bruno

Então, qual código que mostra os alunos em ordem alfabética?

- A) `print(S1, S2, S3, S4, S5)`
- **B) `print(S2, S3, S1, S5, S4)`**
- C) `print(S3, S2, S5, S4, S1)`
- D) `print(S2, S3, S5, S1, S4)`
- E) `print(S2, S3, S5, S4, S1)`

### Questão 2

É muito comum a criação de códigos que tenham algum tipo de interação com o usuário. Uma das maneiras de fazer um código do gênero é utilizando a função `input()`. Essa função por si só aceita qualquer valor de entrada por parte do usuário, mas também é possível especificar o tipo de entrada que será aceita sem erros, por exemplo: `float(input())` quer que a entrada seja um float, `str(input())` quer que a entrada seja um texto, etc.

Suponha que você seja um professor e queira calcular a média de um aluno dispondo de três notas do mesmo. Você define as entradas da seguinte forma para poder digitar as notas:

```
1 s1 = float(input())
2 s2 = float(input())
3 s3 = float(input())
```

Qual deve ser a linha de código para definir a variável “m” como a média aritmética simples das três notas?

- A)  $m = s1 + s2 + s3 / 3$
- **B)  $m = (s1 + s2 + s3) / 3$**
- C)  $m = \text{float}(s1 + s2 + s3 / 3)$
- D)  $m = \text{float}(s1) + \text{float}(s2) + \text{float}(s3) / 3$
- E)  $m = s1 * s2 * s3 / 3$

### Questão 3

Em Python a contagem começa em 0, por exemplo: tem-se uma variável atribuída a = “online” e as letras da variável são a[0] = “o”, a[1] = “n”, etc. O número inteiro entre colchetes é responsável por chamar a parte da variável correspondente.

Quando somamos variáveis do tipo string, resulta-se na escrita de uma outra variável que é resultado da escrita consecutiva da primeira string com a segunda. Veja a seguir um exemplo:

```
1 a = "Alan"
2 b = " "
3 c = "Turing"
4
5 print(a + b + c)
```

Alan Turing

Obs: b corresponde a uma variável do tipo string com texto de um espaço em branco.

Dessa forma, como podemos ter como saída a string “banana” dispondo das seguintes variáveis:

a = “Alan Turing”

b = “Steve Jobs”

- A) `print(a[1] + a[0] + b[2] + a[1] + a[0] + b[2])`
- **B) `print(b[8] + a[2] + a[3] + a[2] + a[3] + a[2])`**
- C) `print(a[3] + a[1] + a[5] + a[1] + a[0] + a[5])`
- D) `print(b[9] + b[1] + a[3] + a[3] + a[1] + a[0])`
- E) `print(a[1] + a[3] + a[5] + a[1] + a[0] + a[5])`

#### Questão 4

As variáveis do tipo lista são um importante artifício de Python para trabalhar com operações matemáticas sem ter que atribuir muitas variáveis.

Imagine que em uma votação para síndico de condomínio tenham 3 candidatos “A”, “B” e “C”. Um código foi responsável por captar os votos dos moradores e teve o seguinte resultado:

```
v = ["A", "B", "A", "C", "C", "A", "C", "C", "B", "A"]
```

Imagine que você queira que uma lista diferente atribua os total de votos para cada candidato com o seguinte código e resultado:

```
1 v = ["A", "B", "A", "C", "C", "A", "C", "C", "B", "A"]
2 r = [["A", 0], ["B", 0], ["C", 0]]
3
4 for i in range(len(v)):
5
6     for j in range(len(r)):
7
8         if v[i] == r[j][0]:
9
10            r[j][1] += 1
11
12 print(r)
```

```
[['A', 4], ['B', 2], ['C', 4]]
```

Com base no código acima, analise as seguintes afirmações e diga quais são corretas:

I - A linha 4 estabelece um laço for que se repete o mesmo número de elementos contidos em v;

II - A linha 6 estabelece um laço for que se repete o mesmo número de elementos contidos em r;

III - O Código compara cada elemento de v com cada elemento de r e, caso sejam iguais, incrementa uma contagem no total de votos em r.

- A) todas.
- B) I, apenas.
- C) II, apenas.
- **D) I e II.**
- E) I e III.

### Questão 5

Na programação, é essencial não confundir **atribuições** com **equações matemáticas tradicionais**. Em Python, o operador = representa **atribuição**, e não uma igualdade matemática.

Considere o seguinte código:

```
1 x = 5
2
3 x = x + 2
4
5 print(x)
```

Qual será a saída exibida pelo programa?

- A) 7
- B) x = 7
- C) 5+2
- D) x + 2
- E) O código apresenta erro, pois  $x = x + 2$  não é uma equação válida.

### Questão 6

Observe o seguinte código e escolha a alternativa que melhor descreve a sua função:

```
1 def v_prog(a0, r, n):
2
3     a_n = a0
4
5     for i in range(n):
6
7         a_n = a_n + r
8
9     return print(a_n)
```

- A) Trata-se de uma progressão geométrica que lista todos os valores da sequência.
- B) Trata-se de uma progressão aritmética que lista todos os valores da sequência.
- C) Trata-se de um programa que prevê o valor do último elemento de uma progressão geométrica com valor inicial  $a_0$ , razão  $r$  e total de elementos  $n$ .
- D) Trata-se de um programa que prevê o valor do último elemento de uma progressão aritmética com valor inicial  $a_0$ , razão  $r$  e total de elementos  $n$ .
- E) Trata-se de um programa que prevê o valor do último elemento de uma progressão aritmética com valor inicial  $a_0$ , razão  $r$  e total de elementos  $n+1$ .

### Questão 7

Observe o seguinte código que estabelece uma sequência de Fibonacci (1,1,2,3,5,8,...):

```
1 def Fib(n):
2     F0 = [1,1]
3
4     for i in range(2,n):
5
6         F0.append(F0[i-2] + F0[i-1])
7
8     return print(F0)
```

Qual será o comportamento da função Fib(n) quando chamada com o argumento n = 6?

- A) A função retorna a sequência [1, 1, 2, 3, 5, 8] e imprime a mesma sequência.
- B) A função retorna e imprime apenas o valor de n.
- C) A função imprime um erro, pois o código contém um problema na lógica de cálculo da sequência.
- D) A função imprime a sequência de Fibonacci até o n-ésimo termo, mas a sequência estará errada.
- E) A função imprime a sequência de Fibonacci até o n-ésimo termo e retorna **None**.

### Questão 8

Observe o código e escolha a alternativa com as afirmações corretas:

```
1 def cont(n):
2
3     a = 0
4
5     while n >= a:
6
7         print(str(a) + "\n")
8
9         a = a + 1
10
11     return
```

I - O código define uma variável a = 0 e, enquanto ela for menor do que o valor de entrada da função "n", exibe o valor de a acrescido de nova linha e realiza o incremento da variável a em uma unidade;

II - A troca do código escrito nas linhas 7 e 9 não alteraria o resultado final do código;

III - A troca da condição "while n >= a" para "while n > a" não causaria mudança no resultado da função se na linha 11 estivesse escrito "return print(str(a))".

- A) I, apenas.
- B) II, apenas.
- C) III, apenas.
- D) Todas.
- E) I e III.

#### Questão 9

Programação é uma excelente maneira de testar proposições lógicas. Observe as proposições a seguir a partir das variáveis declaradas e assinale a alternativa com as afirmações corretas.

```
1 a = 5
2 b = 7.2
3 c = 2.2
```

I - `type(a) == type(c)` : **False**

II - `a == b - c` : **True**

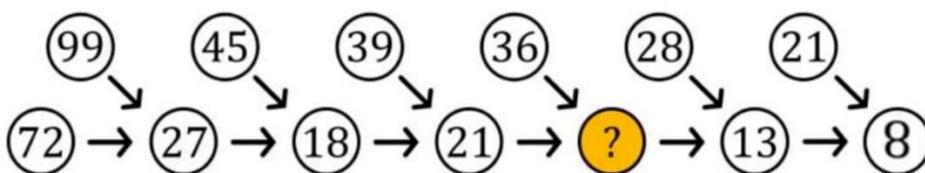
III - `type(a) == type(b-c)` : **False**

São corretas as afirmações:

- A) I, apenas.
- B) I e II.
- C) II e III.
- D) I e III.
- E) Todas.

#### Questão 10

Na primeira fase a seguinte imagem foi apresentada em uma das questões:



Qual dos códigos a seguir descobre a sequência correta?

- A)

```
1 A = [99,45,39,36,28,21]
2
3 x = 72
4
5 for i in range(len(A)):
6
7     x = x + (A[i] - x[i])
8
9 print(x)
```

- B)

```
1 A = [99,45,39,36,28,21]
2
3 x = 72
4
5 for i in range(len(x)):
6
7     x = x + (A[i] - x[i])
8
9 print(x)
```

- C)

```
1 A = [99,45,39,36,28,21]
2
3 x = [72]
4
5 for i in range(len(A)):
6
7     x = x + [(A[i] - x[i])]
8
9 print(x)
```

- D)

```
1 A = [99,45,39,36,28,21]
2
3 x = 72
4
5 for i in range(len(x)):
6
7     x = x + [(A[i] - x[i])]
8
9 print(x)
```

- E)

```
1 A = [99,45,39,36,28,21]
2
3 x = [72]
4
5 for i in range(len(x)):
6
7     x = x + [(A[i] - x[i])]
8
9 print(A)
```