



[www.estudar.com.br](http://www.estudar.com.br)

# Lista de Exercícios

## P2 2017

Programação em Python Poli  
USP





## 1. Tema

Simule o código abaixo. Qual a impressão do programa?

```
def main():
    L0 = []
    v = 32
    while (v > 0):
        v = (v // 2) - 2
        L0.append([v, 2*v])
        L1 = L0
        L1[0][1] = (L1[0][0] + L1[len(L1) - 1][0]) // 2
    print(L0)
main()
```

## 2. Tema

Considere os 4 seguintes trechos de um mesmo programa (T1 até T4) e depois selecione as afirmações verdadeiras sobre eles. As opções sobre cada trecho podem conter desde nenhuma afirmação correta até todas.

<b>T1:</b> <pre>def funcA(L):     v = 0     i = 0     while i &lt; len(L):         if v &lt; L[i]:             v = L[i]         i += 1     return v</pre>	<b>T2:</b> <pre>def funcB(L):     v = L[0]     i = 1     while i &lt; len(L):         if v &gt; L[i]:             v = L[i]         i += 1     return v</pre>
<b>T3:</b> <pre>def funcC(v,L):</pre>	<b>T4:</b> <pre>def main():</pre>



<pre>i = 0 while i &lt; len(L) and v != L[i]:     i += 1 if i &lt; len(L):     R = i else:     R = -1 return R</pre>	<pre>N = [0,1,3,2,4,7,5] A = funcA(N) B = funcB(N) iA = funcC(A,N) iB = funcC(B,N) print(A,B,iA,iB)</pre>
--	---

**T1.**

- A. Retorna o maior valor em  $L$  se contém apenas naturais.
- B. Não retorna o maior valor em  $L$  se contém apenas negativos.
- C. Dá erro de execução se  $L$  for vazia.
- D. Dá erro de execução se  $L$  tiver repetições de valores.

**T2.**

- A. Retorna o menor valor em  $L$  se contém apenas naturais.
- B. Dá erro de execução se  $L$  for vazia.
- C. Não retorna o menor valor em  $L$  se contém apenas negativos.
- D. Dá erro de execução se  $L$  contém valores *float*.

**T3.**

- A. Para todo  $v$  e toda lista  $L$ , retorna a posição do maior valor em  $L$ .
- B. Retorna  $-1$  apenas quando  $L$  é vazia.
- C. Retorna a posição da primeira ocorrência de  $v$  em  $L$ .
- D. Dá erro de execução se  $L$  for vazia.

**T4.**

- A. Mostra a posição do maior natural em  $N$ .
- B. Mostra o maior natural em  $N$ .



C. Determina se  $N$  é crescente.

D. Não mostra a posição do menor natural em  $N$ .

### 3. TEMA

Faça um programa em Python, preenchendo as lacunas (L1 até L12) no código abaixo, que lê uma lista de  $n$  (sendo  $n \geq 1$ ) números reais e a coloca em ordem crescente, mostrando o resultado no final.

```
def encontra_menor(A, p):  
    ''' Devolve o índice do menor valor em A[p], A[p+1],  
        ..., A[len(A)-1] '''  
    L1  
    for L2:  
        if L3:  
            L4  
            achou = True  
    return imenor  
  
def main():  
    n = int(input("Digite n: "))  
    B = []  
    for L5:  
        x = float(input("Digite um número: "))  
        L6  
    L7  
    while L8:  
        m = L9  
        menor = B[m]  
        L10
```



**L11**

**L12**

```
print("Lista ordenada:", L)  
main()
```

Considere os 12 itens seguintes, correspondentes respectivamente a cada uma das 12 lacunas no código acima. Cada item tem 4 opções, selecione aquela que torna o programa acima correto.

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>L1</b>	imenor = 1	imenor = len(A) - 1	imenor = p	imenor = 0
<b>L2</b>	i in range(p+1, len(A) - 1)	i in range(p+1, len(A))	i in range(len(A) - 1)	i in range(len(A))
<b>L3</b>	A[i] < A[p]	A[i] < A[imenor]	A[i] > A[p]	A[i] > A[imenor]
<b>L4</b>	imenor = p	imenor = A[i]	imenor = i	imenor = A[p]
<b>L5</b>	k in range(0, n-1)	k in range(0, n)	k in range(1, n)	k in range(1, n+1)
<b>L6</b>	B + [x]	B[k].append(x)	B.append(x)	B[k] = x
<b>L7</b>	pos = 0	pos = 1	pos = n	pos = n - 1
<b>L8</b>	pos < n	pos >= 0	pos <= n	pos > 0
<b>L9</b>	encontra_menor(B, 0)	encontra_menor(B, pos)	encontra_menor(B, pos + 1)	encontra_menor(B, n)
<b>L10</b>	B[pos] = B[m]	B[m] = pos	B[pos] = menor	B[m] = B[pos]
<b>L11</b>	B[pos] = B[m]	menor = B[pos]	B[m] = B[pos]	B[pos] = menor
<b>L12</b>	pos = m + 1	pos = m - 1	pos = pos + 1	pos = pos - 1

## 4. TEMA



Nesta questão você deve elaborar um programa que, dada uma matriz (uma lista de listas) onde cada linha é uma lista com o nome de um aluno e as notas que ele tirou em cada avaliação, mostre a matriz com uma coluna adicional com a nota média de cada aluno e uma linha adicional com a nota média de cada avaliação. Considere que, na matriz dada, há pelo menos uma linha e pelo menos uma coluna de nota (além da primeira coluna de nome).

Você deve criar e usar duas funções auxiliares, a *mediaPorAluno* e a *mediaPorAvaliacao*, que recebem uma matriz com notas de alunos. A primeira acrescenta na matriz uma coluna com a média das notas de cada aluno, enquanto a segunda acrescenta nela uma nova linha com a nota média de cada coluna de nota. Para elaborar o programa completo, você deve usar apenas trechos de código listados abaixo. Assinale a alternativa que contém os trechos corretos na ordem correta.

Observação: os blocos não estão indentados. Assim, o final do laço é indicado por um '\*' após o número de bloco que contém o último comando do laço. O final de uma função é indicado por '\*\*' após o número de bloco que contém o último comando da função

<pre>#trecho 1 for i in range(numNotas):     somaAluno += aluno[i]</pre>	<pre>#trecho 8 def mediaPorAvaliacao(notasAlunos):     linhaMedias = ['Medias: ']     numAlunos = len(notasAlunos)</pre>	<pre>#trecho 17 somaColuna = 0 col = 0 while col &lt; len(notasAlunos[0]):     col += 1</pre>
<pre>#trecho 2 i = 1 while i &lt; len(aluno):     somaAluno += aluno[i]     i += 1</pre>	<pre>#trecho 9 def main():     alunos = [{" Ana ", 8.5,2.5,4.9},               {" Igor ", 7.3,7.0,6.7},               {" Luiz ", 1.6,2.4,9.2}]</pre>	<pre>#trecho 18 col = 0 while col &lt; len(notasAlunos[0]):     col += 1 somaColuna = 0</pre>
<pre>#trecho 3 for i in range(numNotas - 1):     somaAluno += aluno[i + 1]</pre>	<pre>#trecho 10 def mediaPorAluno(notasAlunos):</pre>	<pre>#trecho 19 for col in range (1, len(notasAlunos[0])):     somaColuna = 0</pre>
<pre>#trecho 4 numNotas = len(notasAlunos)-1</pre>	<pre>#trecho 11 notasAlunos.append(linhaMedias)</pre>	<pre>#trecho 20 mediaAluno = somaAluno/numNotas</pre>



<pre>somaAluno = 0 for aluno in notasAlunos:</pre>	<pre>#trecho 12 notasAlunos[lin + 1] = linhaMedias</pre>	<pre>aluno[i] = mediaAluno</pre>
<pre>#trecho 5 numNotas len(notasAlunos[0]) - 1 for aluno in notasAlunos:     somaAluno = 0</pre>	<pre>#trecho 13 notasAlunos[lin].append(linhaMedias)  #trecho 14 mediaColuna = somaColuna/numAlunos linhaMedias[col] = mediaColuna</pre>	<pre>#trecho 21 mediaAluno = somaAluno/numNotas aluno.append(mediaAluno)</pre>
<pre>#trecho 6 mediaPorAvaliacao(alunos) mediaPorAluno(alunos) print(alunos)</pre>	<pre>#trecho 15 mediaColuna = somaColuna/numAlunos linhaMedias.append(mediaColuna)</pre>	<pre>#trecho 22 lin = 0 while lin &lt; numAlunos:     somaColuna+=notasAlunos[lin][col]     lin += 1</pre>
<pre>#trecho 7 mediaPorAluno(alunos) mediaPorAvaliacao(alunos) print(alunos)</pre>	<pre>#trecho 16 main()</pre>	<pre>#trecho 23 for lin in range(1, numAlunos):     soma += aluno[col]</pre>

Escolha uma alternativa:

- A. 10, 5, 2, \*, 21, \*, \*\*, 8, 19, 22, \*, 15, \*, 11, \*\*, 9, 7, \*\*, 16
- B. 10, 4, 3, \*, 21, \*, \*\*, 8, 17, 23, \*, 14, 11, \*, \*\*, 9, 7, \*\*, 16
- C. 10, 4, 1, \*, \*, 20, \*\*, 8, 17, 23, \*, 15, \*, 12, \*\*, 9, 6, \*\*, 16
- D. 10, 5, 2, \*, 21, \*, \*\*, 8, 19, 22, \*, 15, \*, 11, \*\*, 9, 6, \*\*, 16
- E. 10, 4, 3, \*, 20, \*, \*\*, 8, 18, 22, \*, 15, 12, \*, \*\*, 9, 6, \*\*, 16
- F. 10, 5, 1, \*, 21, \*, \*\*, 8, 17, 22, \*, 13, 11, \*, \*\*, 9, 6, \*\*, 16
- G. 10, 4, 2, \*, 20, \*, \*\*, 8, 17, 23, \*, 14, \*, 11, \*\*, 9, 7, \*\*, 16
- H. 10, 4, 3, \*, 20, \*, \*\*, 8, 18, 22, \*, 13, \*, 12, \*\*, 9, 7, \*\*, 16
- I. 10, 5, 2, \*, \*, 21, \*\*, 8, 19, 23, \*, 15, 11, \*, \*\*, 9, 7, \*\*, 16
- J. 10, 5, 1, \*, 21, \*, \*\*, 8, 18, 23, \*, 13, \*, 11, \*\*, 9, 7, \*\*, 16



## Gabarito

1.  $[[14,7], [5,10], [0,0]]$

2.

T1. Alternativas A e B

T2. Alternativas A e B

T3. Alternativa C

T4. Alternativas A e B

3.

L1. Alternativa C

L2. Alternativa B

L3. Alternativa B

L4. Alternativa C

L5. Alternativa B ou D

L6. Alternativa C

L7. Alternativa A

L8. Alternativa A

L9. Alternativa B

L10. Alternativa D

L11. Alternativa D

L12. Alternativa C

4. Alternativa A