



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

# Lista de Exercícios

P1 2017

Programação em Python Poli

USP





## 1. Tema

Simule o código abaixo. Qual a impressão do programa?

```
def main():
    k = 23
    a = 1
    x = 2
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

## 2. Tema

Considere os 6 seguintes trechos de código (T1 até T6) e depois selecione as afirmações verdadeiras. Considere:

- I. Considere que o usuário sempre digitará uma sequência de naturais entre 1 e 100, terminando com o valor zero (0). Além disso, a sequência (*SEQ*) tem ao menos dois elementos.
- II. As afirmações devem estar corretas quaisquer que sejam as entradas atendendo essas restrições.
- III. Nos trechos de código da tabela abaixo, o termo *SEQ* nas afirmações significa “sequência”.
- IV. Note que a *SEQ* {1,1,1} deve ser considerada crescente e também decrescente.



V. As opções sobre cada trecho podem conter desde nenhuma afirmação correta até todas.

<pre>#Trecho T1 : x = 1 M = 0 while (x != 0):     x = int(input())     if (M &lt; x):         M = x print(M)</pre>	<pre>#Trecho T2 : x = 1 M = 0 while (x != 0):     if (M &lt; x):         M = x     x = int(input()) print(M)</pre>	<pre>#Trecho T3 : x = 1 M = 0 while (x != 0):     x = int(input())     if (M &gt; x):         M = x print(M)</pre>
<pre>#Trecho T4 : x = 1000 M = 1000 while (x != 0):     if (M &gt; x):         M = x     x = int(input()) print(M)</pre>	<pre>#Trecho T5 : x = -1 y = -2 M = 1 while (x != 0):     if (y &gt; x):         M = 0     y = x     x = int(input()) if (M == 1):     print("sim") else:     print("nao")</pre>	<pre>#Trecho T6 : x = -1 y = -2 M = 1 while (x != 0):     if (y &gt; x):         M = 0     else:         M = 1     y = x     x = int(input()) if (M == 1):     print("sim") else:     print("nao")</pre>

**T1.**

- A. Seleciona o menor natural em *SEQ*.
- B. Falha ao tentar selecionar o maior natural em *SEQ*.
- C. Determina se *SEQ* é crescente.



**D.** Seleciona o maior natural em  $SEQ$ .

**T2.**

**A.** Seleciona o maior natural em  $SEQ$ .

**B.** Seleciona o menor natural em  $SEQ$ .

**C.** Falha ao tentar selecionar o menor natural em  $SEQ$ .

**D.** Pode selecionar o menor para algumas  $SEQ$ .

**T3.**

**A.** Falha ao tentar selecionar o menor natural em  $SEQ$ .

**B.** Seleciona o menor natural em  $SEQ$ .

**T4.**

**A.** Falha ao tentar selecionar o menor natural em  $SEQ$ .

**B.** Seleciona o menor natural em  $SEQ$ .

**T5.**

**A.** Imprime sim se  $SEQ$  for crescente.

**B.** Imprime sim se  $SEQ$  for estritamente crescente.

**C.** Imprime sim se  $SEQ$  for decrescente.

**D.** Imprime não se  $SEQ$  for estritamente decrescente.

**T6.**

**A.** Imprime sim se os últimos 2 de  $SEQ$  são estritamente crescentes.

**B.** Imprime não se  $SEQ$  for decrescente.

**C.** Imprime sim se  $SEQ$  for decrescente.

**D.** Imprime não se os últimos 2 de  $SEQ$  não estritamente decrescentes.



### 3. TEMA

Faça um programa em Python, preenchendo as lacunas (L1 até L12) no código abaixo, que remove todos dígitos repetidos de um inteiro  $n$  fornecido, mantendo apenas a primeira ocorrência mais à esquerda de cada dígito:

Exemplos:

$n = 55122345558 \rightarrow$  Saída = 512348

$n = 888778811222 \rightarrow$  Saída = 8712

```
def pertence(d, n):  
    achou = False  
    while L1:  
        L2  
        L3  
        if L4:  
            achou = True  
    return achou  
  
def main():  
    n = int(input("Digite n: "))  
    L5  
    L6  
    while L7:  
        L8  
        L9
```



```
if L10:  
    L11  
    L12  
    print(saída)  
main()
```

Considere os 12 itens seguintes, correspondentes respectivamente a cada uma das 12 lacunas no código acima. Cada item tem 5 opções, selecione aquela que torna o programa acima correto.

	A	B	C	D	E
L1	<code>d &lt;= n</code>	<code>n &gt;= 0</code>	<code>n != 0 and achou</code>	<code>d &lt; n</code>	<code>n &gt; 0 and achou == False</code>
L2	<code>r = n%10</code>	<code>n = n//10</code>	<code>n = n%10</code>	<code>r = n//10</code>	<code>d = d + 1</code>
L3	<code>n = n%10</code>	<code>n = n//10</code>	<code>d = d + 1</code>	<code>r = n%10</code>	<code>r = n//10</code>
L4	<code>d == n</code>	<code>r != d</code>	<code>d != n</code>	<code>d != n%10</code>	<code>r == d</code>
L5	<code>saída = 0</code>	<code>saída = n//100</code>	<code>saída = d//10</code>	<code>saída = n</code>	<code>saída = 1</code>
L6	<code>pot = n*10</code>	<code>pot = 10**n</code>	<code>pot = 100</code>	<code>pot = 1</code>	<code>pot = 10</code>
L7	<code>n &gt;= 10</code>	<code>n &gt; 10</code>	<code>n &gt; 0</code>	<code>n &gt;= 0</code>	<code>pot &lt;= n</code>
L8	<code>d = n//10</code>	<code>r = n%10</code>	<code>n = n%10</code>	<code>n = n//2</code>	<code>r = n//10</code>
L9	<code>r = n//2</code>	<code>r = n%10</code>	<code>n = d//10</code>	<code>n = n%10</code>	<code>n = n//1-</code>
L10	<code>pertence (n, r)</code>	<code>pertence (r,n//10)</code>	<code>pertence (r, n)</code>	<code>pertence (n, r) == False</code>	<code>pertence (r, n) == False</code>
L11	<code>saida = saida*pot + r</code>	<code>saida += n//pot</code>	<code>saida = saida + r*pot</code>	<code>saida += r</code>	<code>saida = saida*pot+r</code>



<b>L12</b>	pot = pot*10	pot = pot*100	pot = 10**n	pot = pot//10	pot = pot+1
------------	-----------------	---------------	-------------	------------------	-------------

#### 4. TEMA

Nesta questão você deve elaborar um programa que resolva o seguinte problema: dada uma sequência de  $n$  números ( $n \geq 2$ ), diga se o valor das diferenças entre cada dois números consecutivos é uma sequência ( $S$ ) estritamente crescente.

Exemplo:

Entrada:  $n = 4, S = 8, 9, 10, 11$

Saída: 'diferencas nao formam seq. crescente'

Para isso você deve usar apenas os trechos de código indicados abaixo. Assinale a alternativa que contém os blocos corretos na ordem correta.

Atenção: 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.

Observação 1: As variáveis do programa são apenas *quantidade*; *anterior*; *atual*; *contador*; *diferenca\_atual*; *diferenca\_anterior*.

Observação 2: Não tente usar todas as combinações, tente codificar o programa e depois escolha os trechos adequados.



<pre>#trecho 1 diferenca_anterior = atual - anterior</pre>	<pre>#trecho 10 contador = contador + 1</pre>	<pre>#trecho 17 atual = int(input("Digite um número:")) anterior = atual diferenca_atual = atual - anterior</pre>
<pre>#trecho 2 crescente = False</pre>	<pre>#trecho 11 if(diferenca_atual&lt;diferenca_anterior):</pre>	<pre>#trecho 18 anterior = atual atual = int(input("Digite um número:")) diferenca_atual = atual - anterior</pre>
<pre>#trecho 3 crescente = True contador = 2</pre>	<pre>#trecho 12 if(diferenca_atual&lt;=diferenca_anterior):</pre>	
<pre>#trecho 4 diferenca_atual = diferenca_anterior</pre>	<pre>#trecho 13 while (contador &lt;= quantidade):</pre>	<pre>#trecho 19 atual = int(input("Digite um número:")) anterior = atual diferenca_atual = atual - anterior</pre>
<pre>#trecho 5 contador = contador + 2</pre>	<pre>#trecho 14 while (contador &lt; quantidade):</pre>	
<pre>#trecho 6 diferenca_anterior = diferenca_atual</pre>	<pre>#trecho 15 quantidade = int("tamanho da seq.:") anterior = int("Digite um número:") atual = int("Digite um número:")</pre>	<pre>#trecho 20 else:     crescente = True</pre>
<pre>#trecho 7 crescente = 0</pre>		<pre>#trecho 21 if ((diferenca_atual - diferenca_anterior)&gt;0):</pre>
<pre>#trecho 8 crescente = True contador = 0</pre>	<pre>#trecho 16 quantidade = int(input("tamanho da seq.:")) anterior = int(input("Digite um número:")) atual = int(input("Digite um número:"))</pre>	<pre>#trecho 22 if (crescente):     print("Diferenças formam seq. estr. \ crescente") else:     print("Diferenças não formam seq. \ estr. crescente")</pre>
<pre>#trecho 9 crescente = 1 contador = 0</pre>		

Escolha uma alternativa:

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





I. 15, 1, 8, 13, 6, 17, 11, 2, 20, 10, \*, 22

J. 16, 1, 13, 6, 19, 11, 2, 10, \*, 22

K. 15, 1, 8, 13, 6, 17, 11, 2, 20, 10, \*, 22

## Gabarito

1. 14

35

28

35

7

2.

T1. Alternativa D

T2. Alternativas A e D

T3. Alternativa A

T4. Alternativa B

T5. Alternativas A, B e D

T6. Alternativas A e D

3.

L1. Alternativa E

L2. Alternativa A

L3. Alternativa B

L4. Alternativa E

L5. Alternativa A

L6. Alternativa D

L7. Alternativa C

L8. Alternativa B

L9. Alternativa E

L10. Alternativa E



**L11.** Alternativa C

**L12.** Alternativa A

**4.** Alternativa F