



www.estudar.com.br

Introdução à Computação em Python

Prova 2 Poli USP - 2017





1. (adaptado)

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. Considere os 6 seguintes trechos de código (T1 até T6) e depois selecione as afirmações verdadeiras. **Considere:** 1. 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 2



elementos. **2.** As afirmações devem estar corretas quaisquer que sejam as entradas atendendo essas restrições. **3.** Nos trechos de código da tabela abaixo, o termo **SEQ** nas afirmações significa “sequência”. **4.** Note que a SEQ {1,1,1} deve ser considerada **crecente** e também **decrecente**. **5.** 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<x): M = x print(M)</pre>	<pre>#Trecho T2 : x = 1 M = 0 while (x != 0): if (M<x): M = x x = int(input()) print(M)</pre>	<pre>#Trecho T3 : x = 1 M = 0 while (x != 0): x = int(input()) if (M>x): M = x print(M)</pre>
<pre>#Trecho T4 : x = 1000 M = 1000 while (x!=0): if (M>x): M = x x = int(input()) print(M)</pre>	<pre>#Trecho T5 : x = -1 y = -2 M = 1 while (x!= 0): if (y>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>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 a 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 a SEQ for crescente
- b. Imprime sim se a SEQ for estritamente crescente
- c. Imprime sim se a SEQ for decrescente
- d. Imprime não se a SEQ for estritamente decrescente

T6.

- a. Imprime sim se os últimos 2 da SEQ são estritamente crescentes
- b. Imprime não se a SEQ for decrescente
- c. Imprime sim se a SEQ for decrescente
- d. Imprime não se os últimos 2 da SEQ não estritamente decrescentes

3. 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 a esquerda de cada dígito:

Exemplos:

$n = 55122345558 \rightarrow \text{saida} = 512348$, $n = 888778811222 \rightarrow \text{saida} = 8712$

```
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.

	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>
<u>L1</u>	$d \leq n$	$n \geq 0$	$n \neq 0$ and achou	$d < n$	$n > 0$ and achou == False
<u>L2</u>	$r = n \% 10$	$n = n // 10$	$n = n \% 10$	$r = n // 10$	$d = d + 1$



<u>L3</u>	<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>
<u>L4</u>	<code>d == n</code>	<code>r != d</code>	<code>d != n</code>	<code>d != n%10</code>	<code>r == d</code>
<u>L5</u>	<code>saída = 0</code>	<code>saída = n//100</code>	<code>saída = d//10</code>	<code>saída = n</code>	<code>saída = 1</code>
<u>L6</u>	<code>pot = n*10</code>	<code>pot = 10**n</code>	<code>pot = 100</code>	<code>pot = 1</code>	<code>pot = 10</code>
<u>L7</u>	<code>n >= 10</code>	<code>n > 10</code>	<code>n > 0</code>	<code>n >= 0</code>	<code>pot <= n</code>
<u>L8</u>	<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>
<u>L9</u>	<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>
<u>L10</u>	<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>
<u>L11</u>	<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>
<u>L12</u>	<code>pot = pot*10</code>	<code>pot = pot*100</code>	<code>pot = 10**n</code>	<code>pot = pot//10</code>	<code>pot = pot+1</code>

4. 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. **Exemplos:** $n = 4$, $S = 8, 9, 10, 11 \rightarrow$ 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 identados assim o final do laço é indicado por um '*', após o número de bloco que contém o último comando do laço. **DICA 1: As variáveis do programa são APENAS:** quantidade; anterior; atual; contador; diferença_atual; diferença_anterior.



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

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

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. d

T2. a e d

T3. a

T4. b

T5. a, b e d

T6. a e d

3.

L1. e

L2. a

L3. b

L4. e

L5. a

L6. d

L7. c

L8. b

L9. e

L10. e

L11. c

L12. a

4. F