



www.estudar.com.br

Lista de Exercícios

P1 2018

Programação em Python Poli

USP





1. Tema

Simule o código abaixo e selecione as opções correspondentes à saída impressa do programa:

```
def main():
    k = 25
    a = 0
    while k >= a:
        b = k // 2
        if b * 2 == k:
            a = a + 3
        else:
            a = a + 2
        k = k - 3
        x = a % 7
        print(x * 7 + 14)
main()
```

a. O primeiro número impresso é:

A. 35 **B.** 56 **C.** 21 **D.** 42 **E.** 28 **G.** 14 **H.** 49

b. O segundo número impresso é:

A. 14 **B.** 28 **C.** 49 **D.** 21 **E.** 42 **G.** 35 **H.** 56

c. O terceiro número impresso é:



A. 49 B. 35 C. 42 D. 14 E. 28 G. N/A H. 56 I. 21

d. O quarto número impresso é:

A. 21 B. 14 C. 49 D. N/A E. 35 G. 56 H. 28 I. 42

e. O quinto número impresso é:

A. 14 B. N/A C. 35 D. 49 E. 28 G. 42 H. 21 I. 56

2. Tema

Para cada item envolvendo a análise de expressões lógicas, assinale as opções corretas pintando as quadrículas. As opções sobre cada item podem conter desde nenhuma opção correta até todas.

a. Dadas duas variáveis inteiras a e b , selecione todas as expressões equivalentes a $a \geq b$:

A. $a < b == False$

B. $not (a \leq b \text{ or } a \neq b)$

C. $(a//2) \geq (b//2)$

D. $a > b + 1$

E. $not (a < b)$

F. $a > b \text{ or } a == b$

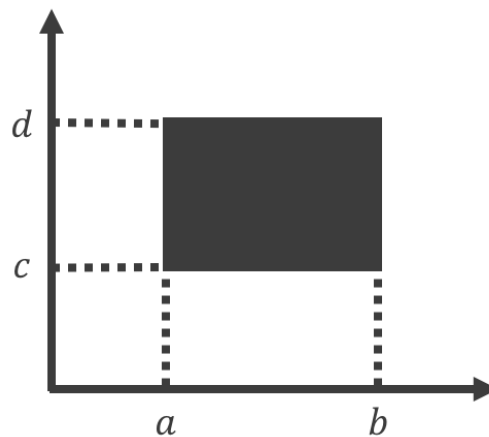
G. $a > b \text{ and } a == b$

H. $not (a \leq b \text{ and } a \neq b)$

I. $a > b - 1$

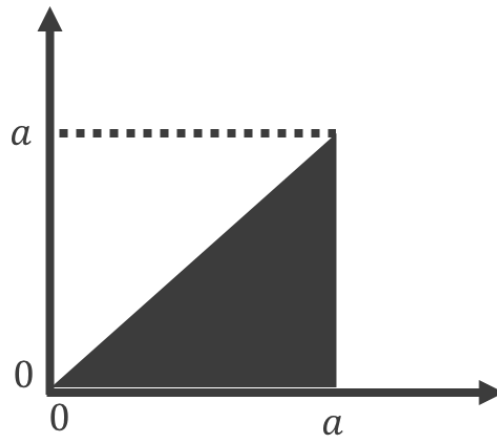


b. Dadas as coordenadas reais x e y de um ponto, selecione todas expressões que geram *True* se esse ponto está na região sombreada da figura abaixo e *False* caso contrário. A região sombreada não inclui as linhas de fronteira.



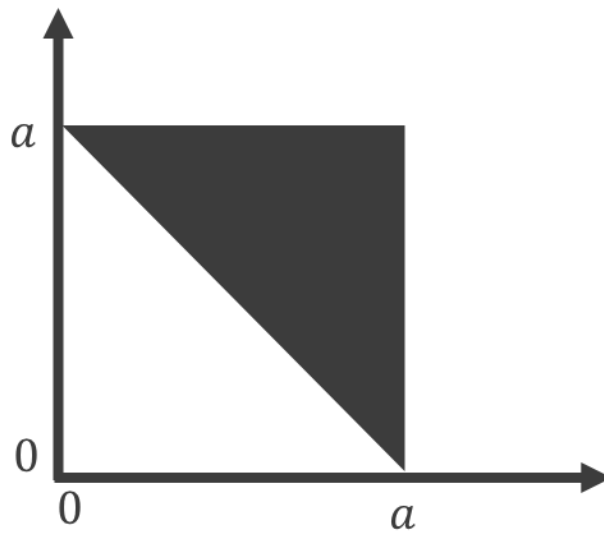
- A.** $\text{not } (x \leq a \text{ or } x \geq b) \text{ and not } (y \leq c \text{ or } y \geq d)$
- B.** $x > a \text{ and } x < b \text{ and } y > c \text{ and } y < d$
- C.** $(x < b \text{ and } y < d) \text{ or not } (x \leq a \text{ and } y \leq c)$
- D.** $\text{not } (x \leq a \text{ and } x \geq b \text{ and } y \leq c \text{ and } y \geq d)$
- E.** $x > a \text{ or } x < b \text{ or } y < c \text{ or } y < d$
- F.** $(x < b \text{ and } y < d) \text{ and not } (x \leq a \text{ or } y \leq c)$
- G.** $\text{not } (x \leq a \text{ and } x \geq b) \text{ or not } (y \leq c \text{ and } y \geq d)$
- H.** $\text{not } (x \leq a \text{ or } x \geq b \text{ or } y \leq c \text{ or } y \geq d)$

c. Dadas as coordenadas reais x e y de um ponto, selecione todas expressões que geram *True* se esse ponto está na região sombreada da figura abaixo e *False* caso contrário. A região sombreada não inclui as linhas de fronteira.



- A.** $x < a$ and $y > 0$ and $x > y$
- B.** $\text{not } (x \geq a \text{ and } y \leq 0 \text{ and } x \leq y)$
- C.** $x < a$ or $y > 0$ or $x > y$
- D.** $y < x$ and $\text{not } (x \geq a \text{ or } 0 \geq y)$
- E.** $x < a$ or $(0 < y \text{ and } y < x)$
- F.** $\text{not } (x \geq a \text{ or } y \leq 0 \text{ or } x \leq y)$
- G.** $x > 0$ or $x < a$ or $y < a$ or $y > 0$ or $x > y$
- H.** $x \geq a$ or $y \leq 0$ or $x \leq y$
- I.** $x < a$ and $y > 0$ and $\text{not } (x \leq y)$
- J.** $x > 0$ and $x < a$ and $y < a$ and $y > 0$ and $x > y$

d. Dadas as coordenadas reais x e y de um ponto, selecione todas expressões que geram *True* se esse ponto está na região sombreada da figura abaixo e *False* caso contrário. A região sombreada não inclui as linhas de fronteira.



- A. $(x \leq y \text{ and } y < a) \text{ and } x + y > a$
- B. $x < a \text{ and } y < a \text{ and } x - y < a$
- C. $(x \leq y \text{ and } y < a) \text{ and } x + y < a$
- D. $x < a \text{ and } y < a \text{ and } x + y < a$
- E. $\text{not } (x \geq a \text{ and } y \geq a \text{ and } x + y \leq a)$
- F. $x < a \text{ and } y < a \text{ and } x + y < a$
- G. $x < a \text{ or } y < a \text{ or } x + y > a$
- H. $x < a \text{ or } y < a \text{ or } x + y < a$
- I. $x \leq a \text{ and } y \leq a \text{ and } x > a \cdot y$
- J. $\text{not } (x \geq a \text{ or } y \geq a \text{ or } x + y \leq a)$
- K. $\text{not } (x \geq a \text{ or } y \geq a) \text{ and } x > a - y$
- L. $x < a \text{ or } y < a \text{ or } x - y < a$
- M. $x < a \text{ and } y < a \text{ and } x + y == a$
- N. $x < a \text{ or } y < a \text{ or } x + y == a$

3. TEMA



Três números inteiros positivos a , b e c , com $a < b < c$, formam um trio Pitagórico se $a^2 + b^2 = c^2$. Por exemplo, os números 3, 4 e 5 formam um trio Pitagórico pois $3^2 + 4^2 = 5^2$. Alguns números inteiros positivos podem ser escritos como a soma de um trio Pitagórico. Por exemplo, 12 é um desses números pois $3 + 4 + 5 = 12$. Preencha as lacunas no código abaixo (L1 até L12), de forma a obter um programa que lê um número inteiro n ($n > 0$) e verifica se ele corresponde à soma de um trio Pitagórico. Em caso afirmativo, o programa deve imprimir os valores do trio e, caso contrário, deve imprimir que o número não é soma de trio Pitagórico. Por exemplo, $n = 12$ é soma de trio Pitagórico (3,4,5) enquanto $n = 10$ não é soma de trio Pitagórico.

```
def main():
    n = int(input("Entre com n:"))
    L1
    L2
    L3
        L4
        L5
            L6
            L7
                L8
                L9
                    L10
                    L11
    L12
        Print("%d não eh soma de trio Pitagórico" (%n))
main()
```



Para cada um dos 12 itens a seguir, correspondendo as lacunas no código acima, assinale a única resposta que torna o programa acima correto.

L1:

A. $c = 1$

B. $b = 1$

C. $a = b = c = n$

D. $a = 1$

E. $a = b = c = n - 1$

L2:

A. $achei = False$

B. $achei = True$

C. $c = a + 1$

D. $c = b + 1$

E. $b = a + 1$

L3:

A. *while* $a < n$ and *achei*:

B. *while* $a < n$ and not *achei*:

C. *while* not *achei*:

D. *while* *achei*:

E. *while* $a \leq n$ and *achei*:

L4:

A. $c = c + 1$

B. $b = b + 1$

C. $b = a + 1$



D. $a = a + 1$

E. $c = b + 1$

L5:

A. *while* *achei*:

B. *while not* *achei*:

C. *while* $b \leq n - a$ *and* *achei*:

D. *while* $b < n - a$ *and not* *achei*:

E. *while* $b < n - 1$ *and* *achei*:

F. *while* $b < n$ *and* *achei*:

L6:

A. $a = n - b - c$

B. $c = n - a - b$

C. $b += 1$

D. $c += 1$

E. $a += 1$

L7:

A. *if* $a + b + c == 1$:

B. *if* $a*a + b*b != c*c$:

C. *if* $a + b + c != n$:

D. *if* $a + b + c == n$:

E. *if* $a*a + b*b == c*c$:

L8:

A. $c += 1$

B. `print("%d eh soma de trio Pitagórico (%d, %d, %d)" %(n, a, b, c))`



C. $a += 1$

D. `print(%d nao eh soma de trio Pitagórico%(n))`

E. $b += 1$

L9:

A. `achei = False`

B. `achei = True`

C. $a = b + 1$

D. $b = c + 1$

E. $n = a + b + c$

L10:

A. $a = b + 1$

B. $c += 1$

C. $a += 1$

D. $b += 1$

E. $b = c + 1$

L11:

A. $a = b + 1$

B. $b += 1$

C. $c += 1$

D. $b = c + 1$

E. $a += 1$

L12:

A. `if a + b + c == n and a*a + b*b == c*c:`

B. `while not achei:`



C. *while* *achei*:

D. *if not* *achei*:

E. *if* $a*a + b*b \neq c*c$:

4. TEMA

Nesta questão, você deve elaborar um programa que, dados n e uma sequência de n números, imprima, para cada um dos números, a soma de seus dígitos em posições pares (segundo, quarto, sexto, etc.) contadas a partir do dígito menos significativo e a soma de seus dígitos em posições ímpares (primeiro, terceiro, quinto, etc.). Por exemplo, para o número 4897, a soma dos dígitos em posições pares é $9 + 4 = 13$ e a soma dos dígitos em posições ímpares é $7 + 8 = 15$. Ao final, o programa deve também indicar quantos números possuem a primeira soma estritamente maior que a segunda e quantos têm a segunda soma estritamente maior que a primeira. Considere o exemplo abaixo:

Entrada	Saída
4	pares: 7
274	ímpares: 6
4897	pares: 13
4444	ímpares: 15
5	pares: 8
	ímpares: 8
	pares: 0
	ímpares: 5
	1, 2



As variáveis do programa são apenas: n (quantidade de números), s_{pr_m} e s_{im_m} (contadores), dig (um dígito), s_{d_pr} e s_{d_im} (somadas), x (um número), d_{im} (uma variável booleana) e i (um contador).

Assinale a única alternativa que contém os blocos corretos (dentro os abaixo) na ordem correta.

#trecho 1 <code>x = int(input(""))</code>	#trecho 10 <code>s_d_pr = 1 s_d_im = 1</code>	#trecho 18 <code>j = 0 while (j < x):</code>
#trecho 2 <code>n = int(input("")) i = 1 q_s_pr = 0 q_s_im = 0</code>	#trecho 11 <code>eh_im = True</code>	#trecho 19 <code>if (s_d_pr > s_d_im): q_s_pr += 1 else: q_s_im += 1</code>
#trecho 3 <code>n = int(input("")) i = 0 q_s_pr = 1 q_s_im = 1</code>	#trecho 12 <code>eh_im = not eh_im</code>	
#trecho 4 <code>n = int(input("")) i = 0 q_s_pr = 0 q_s_im = 0</code>	#trecho 13 <code>eh_im += 1</code>	#trecho 20 <code>if (s_d_pr > s_d_im): q_s_pr += 1 elif (s_d_pr < s_d_im): q_s_im += 1</code>
#trecho 5 <code>x = 0</code>	#trecho 14 <code>i = i + 1</code>	
#trecho 6 <code>dig = x % 10 x = x / 10</code>	#trecho 15 <code>while (i <= n+1):</code>	#trecho 21 <code>if (eh_im): s_d_im += dig else: s_d_pr += dig</code>
#trecho 7 <code>dig = x % 10 x = x // 10</code>		#trecho 22 <code>print(q_s_pr,",", q_s_im)</code>
#trecho 8 <code>x = x // 10 dig = x % 10</code>	#trecho 16 <code>while (i < n):</code>	#trecho 23 <code>print("pares:", s_d_pr) print("impares:", s_d_im)</code>
#trecho 9 <code>s_d_pr = 0 s_d_im = 0</code>	#trecho 17 <code>while (x != 0):</code>	

A. 4, 16, 1, 9, 11, 17, 7, 21, 23, 20, 14, 22

B. 3, 3, 18, 23, 9, 18, 21, 19, 14, 22



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



Gabarito

1.

- a. Alternativa E
- b. Alternativa C
- c. Alternativa D
- d. Alternativa E
- e. Alternativa D

2.

- a. Alternativas A, E, F, H e I
- b. Alternativas A, B, F e H
- c. Alternativas A, D, F, I e J
- d. Alternativas D, J e K

3.

- L1.** Alternativa D
- L2.** Alternativa A
- L3.** Alternativa B
- L4.** Alternativa C
- L5.** Alternativa D
- L6.** Alternativa B
- L7.** Alternativa E
- L8.** Alternativa B
- L9.** Alternativa B
- L10.** Alternativa D
- L11.** Alternativa E
- L12.** Alternativa D



4. Alternativa F.