

Prova 2

02/12/2017

08:00 –10:00h

Matrícula: \_\_\_\_\_ Nome: \_\_\_\_\_

Marque com um X sua turma **TEÓRICA**:

	<b>Turma</b>	<b>Dia da Semana</b>	<b>Horário</b>	<b>Professor</b>
	1	Quinta-feira	10-12	Mauro
	2	Sexta-feira	10-12	Jean H.
	3	Sexta-feira	08-10	Thales
	4	Sexta-feira	14-16	Thales
	5	Quinta-feira	14-16	Lucas

**OBSERVAÇÕES:**

- As questões podem ser resolvidas a lápis.
- Entende-se por algoritmo refinado completo um algoritmo contendo a representação do código em português, de forma clara, não ambígua, de modo que cada instrução do algoritmo possa ser traduzida em uma instrução da linguagem de programação.
- Para a leitura de dados, basta usar instruções em formato algorítmico como “**Leia n**”, “**Leia A, B, C**” etc. Não precisa se preocupar com as mensagens escritas antes dessas leituras.

A tabela abaixo apresenta a ordem de prioridade dos operadores e comandos Python mais comuns. Essa informação pode ser útil para você na resolução das questões da prova.

<b>Prioridade</b>	<b>Operador(es) e comando =</b>	<b>Exemplo</b>
1	- (unário)	-x
2	**	x ** 3
3	*    /    //    %	x / y
4	+    -	x - y
5	<    <=    >    >=    ==    !=	x < y
6	not	not x > 0
7	and	x < y and x > 0
8	or	x < y or x == 0
9	=	x = 2

**(6 pontos)**

```
import numpy as np

def montaMatriz( m, n ):
    x = np.empty( (m,n), dtype=int )
    k = 0
    for i in range( 0, m ):
        for j in range( 0, n ):
            x[i][j] = k
            k = k + 1
    return x

def s1Matriz( x, i ):
    m, n = x.shape # Pega número de linhas e colunas da matriz
    a = 0
    for j in range( 0, n ):
        a = a + x[i][j]
    return a

def s2Matriz( x, j ):
    m, n = x.shape # Pega número de linhas e colunas da matriz
    a = 0
    for i in range( 0, m ):
        a = a + x[i][j]
    return a

A = montaMatriz( 2, 2 )
for i in range( 0, 2 ):
    print( s1Matriz( A, i ))
for j in range( 0, 2 ):
    print( s2Matriz( A, j ))
```

Tela do Computador	Rascunho (rastreamento das variáveis)

**Questão 2****(10 pontos)**

O MDC – Máximo Divisor Comum – entre dois números inteiros A e B é o maior número inteiro que é divisor tanto de A quanto de B. Exemplos:

A	B	MDC( A, B )
20	30	10
30	3	3
99	77	11
7	9	1

O programa a seguir faz a leitura de dois números inteiros pelo teclado e imprima o MDC entre os dois. Você deve completar o código da função mdc() para que ele funcione corretamente.

```
def mdc( x, y ):
```

```
def programa_principal():  
    a = int( input('A = '))  
    b = int( input('B = '))  
    print( mdc(a, b) )
```

```
programa_principal()
```

### Questão 3

(12 pontos)

Em um determinado concurso, vários candidatos fizeram uma prova de múltipla escolha. O edital do concurso previa que a vaga seria preenchida pelo candidato que obtivesse a maior nota. Mas o estagiário que redigiu o edital se esqueceu de incluir a regra de desempate! Agora que o concurso já foi realizado, os responsáveis precisam saber o tamanho do problema que tem em mãos... descobrir quantas pessoas empataram com a maior nota!

Dentro do programa você já dispõe do código (100% correto) que lê o arranjo **notas** contendo a nota de cada candidato, e determina a variável **n** contendo o número de valores dentro do arranjo **notas** (veja abaixo).

Em seguida, o programa deve imprimir dois valores como resposta: a maior nota e quantos candidatos obtiveram essa nota. Esses valores podem ser escritos em uma única linha, ou um em cada linha.

Exemplo 1:

Entre com as notas separadas por espaço:

20 7 9 13 2 12 23

Resultado: 23 1

Exemplo 2:

Entre com as notas separadas por espaço:

7 5 2 4 6 8 9 4 9 2

Resultado: 9 2

Exemplo 3:

Entre com as notas separadas por espaço:

5 5 5 5 5

Resultado: 5 5

### SOLUÇÃO:

```
# Leitura do vetor de notas e cálculo de n:
sv = input('Entre com as notas separadas por espaço:\n')
notas = np.empty( 0, dtype=int )
for x in sv.split():
    notas = np.append( notas, int(x) )
n = len( notas )

# Determinação do resultado:
```

✓

#### Questão 4

(7 pontos)

Em uma aula prática, foi dada a seguinte tarefa:

Escreva um programa que faça o seguinte:

1. Lê um valor inteiro n pelo teclado e cria uma matriz n x n de valores aleatórios;
2. Calcula e escreve na tela a média e o desvio padrão dos valores da matriz, além da média - desvio padrão e a média + desvio padrão;
3. Calcula e escreve na tela uma matriz C, tal que  $C_{ij} = -1$  se  $A_{ij} < (\text{Média} - \text{DP})$ ,  $C_{ij} = 1$  se  $A_{ij} > (\text{Média} + \text{DP})$ , e  $C_{ij} = 0$  caso contrário.

onde: 
$$DP = \sqrt{\frac{\sum_i \sum_j (x_{ij} - \bar{x})^2}{n^2}}$$
 sendo  $\bar{x}$  a média dos valores.

E de acordo com o enunciado, o programa deveria fornecer a seguinte saída:

Entre com a Ordem da matriz: 4

Matriz A:

11.53	15.02	12.66	11.44
8.90	13.56	9.19	18.73
20.24	8.05	16.63	11.11
11.93	19.44	1.49	1.83

Média: 11.98

Desvio Padrão: 5.30

Média - DP: 6.68

Média + DP: 17.29

Matriz C:

0	0	0	0
0	0	0	1
1	0	0	0
0	1	-1	-1

No entanto, após um aluno implementar e executar o programa, ele obteve a seguinte saída:

Entre com a Ordem da matriz: 4

Matriz A:

11.53	15.02	12.66	11.44
8.90	13.56	9.19	18.73
20.24	8.05	16.63	11.11
11.93	19.44	1.49	1.83

Média: 11.98

Desvio Padrão: 6.33

Média - DP: 5.65

Média + DP: 18.32

Matriz C:

0	0	0	0
0	0	0	1
1	0	0	0
0	1	0	0

O aluno então observou os seguintes problemas:

1. O valor da média está correto, mas o desvio padrão não está;
2. Na montagem da matriz C, alguns valores que deveriam ser -1 estão zerados.

Abaixo você pode ver o código implementado pelo aluno. Descreva quais foram os erros cometidos e como eles podem ser corrigidos. Use os números das linhas para facilitar a sua resposta.

```
1  import numpy as np
2
3  n = int( input('Entre com a Ordem da matriz: '))
4  np.random.seed( 0 )
5  A = np.random.uniform( 0, 21, (n,n) )
6
7  print('\nMatriz A:')
8  for i in range( 0, n ):
9      for j in range( 0, n ):
10         print('%8.2f' % A[i][j], end='')
11     print()
12
13     soma = 0
14     for i in range( 0, n ):
15         for j in range( 0, n ):
16             soma = soma + A[i][j]
17     media = soma / n**2
18
19     for i in range( 0, n ):
20         for j in range( 0, n ):
21             soma = soma + (A[i][j] - media) ** 2
22     desvpad = (soma / n**2) ** 0.5
23
24     print()
25     print('Média:          %5.2f' % media )
26     print('Desvio Padrão: %5.2f' % desvpad )
27     print('Média - DP:     %5.2f' % (media - desvpad))
28     print('Média + DP:     %5.2f' % (media + desvpad))
29
30     C = np.empty( (n,n), dtype=int )
31     for i in range( 0, n ):
32         for j in range( 0, n ):
33             a = A[i][j]
34             if a < media-desvpad:
35                 C[i][j] = -1
36             if a > media+desvpad:
37                 C[i][j] = 1
38             else:
39                 C[i][j] = 0
40
41     print('\nMatriz C:')
42     for i in range( 0, n ):
43         for j in range( 0, n ):
44             print('%8d' % C[i][j], end='')
45     print()
```

**SOLUÇÃO:**