

INSTRUÇÕES GERAIS

1. A avaliação é individual e sem consulta.
 2. A única linguagem de programação aceita é Python 3.
 3. Só é permitido o uso de folhas fornecidas pelo professor.
 4. Questões sem desenvolvimento são consideradas incorretas.
 5. As questões podem apresentar restrições em seu enunciado. Será atribuída nota ZERO para a questão caso alguma de suas restrições não seja cumprida.
 6. Os programas escritos devem funcionar para qualquer caso de entrada possível definido pelo enunciado, não apenas para o(s) exemplo(s) dado(s).
 7. O(A) estudante não pode sair da sala durante a avaliação. Caso o faça, então deve encerrar a prova e entregar as soluções produzidas até o momento.
 8. Conversas e trocas de qualquer tipo de mensagem verbal ou textual são proibidas durante a avaliação e são consideradas tentativa de “cola”.
 9. É proibido o uso de smartphone, tablet, notebook ou qualquer outros dispositivos eletrônicos durante a prova, sendo o acesso aos mesmos considerado tentativa de “cola”.
 10. Será atribuída nota ZERO aos estudantes que: (i) forem pegos tentando “colar”; (ii) forem pegos tentando “passar cola”; (iii) que tenham “colado”, ou (iv) que tenham “passado cola”. A coordenação do curso será notificada do ocorrido para tomar as providências cabíveis.
-

QUESTÃO 1 - O Computador Ainda Sou Eu!

a) Realize o teste de mesa do programa apresentado e escreva exatamente o que ele emite na saída padrão. A formatação de saída será considerada na avaliação.

```
1  def teste(v, tam):
2      for k in range(tam):
3          v[k] = 0
4
5  vet = [-5, 3, 2, 9, 2, 2]
6  j = 4
7
8  valores = 0
9  for i in range(j):
10     if vet[i] == 2:
11         valores = valores + 2
12
13 print(valores)
14
15 for i in range(6):
16     vet[i] = j
17     j = j - 1
18
19 for i in range(len(vet)):
20     print("%i " % vet[i], end="")
21 print()
22
23 teste(vet, 3)
24 for i in range(6):
25     print("%i " % vet[i], end="")
```

b) Realize o teste de mesa do programa apresentado e escreva exatamente o que ele emite na saída padrão. A formatação de saída será considerada na avaliação.

```
1 def funcao1(x, y):
2     return y / (x + 1)
3
4 def funcao2(x, y):
5     z = x + 2
6     y = z
7     w = y
8     return x, y
9
10 c, b = 0, 5
11 c, b = funcao2(c, b)
12 print(c, b)
13 print("%1.4f" % funcao1(c, b))
```

Distribuição de Pontos

Esta questão vale 2,0 dos 10,0 pontos da avaliação. As partes (a) e (b) valem 1,0 ponto cada.

QUESTÃO 2 - Matriz Diagonal

Uma matriz diagonal, em álgebra linear, é uma matriz cujos elementos exteriores à diagonal principal são iguais a zero. Uma matriz diagonal deve ser quadrada, também.

Escreva um programa que lê o tamanho da matriz e, em seguida, seu conteúdo. Como resultado seu programa deve indicar se a matriz é ou não é diagonal.

Entrada

A primeira linha da entrada é composta por dois números inteiros não negativos, $1 \leq L \leq 1000$ e $1 \leq C \leq 1000$, onde L é a quantidade de linhas e C a quantidade de colunas da matriz. As próximas L linhas são compostas, cada uma, por C valores reais separados por um espaço em branco. Cada uma desses valores corresponde a um dos coeficientes da matriz na linha atual.

Saída

O programa deve imprimir como saída o texto "A matriz é diagonal" caso a matriz informada seja diagonal ou "A matriz não é diagonal", caso contrário.

Exemplo

Entrada	Saída
4 4 1.0 0.0 0.0 0.0 0.0 9.3 0.0 0.0 0.0 0.0 -8.0 0.0 0.0 0.0 0.0 7.0	A matriz é diagonal
Entrada	Saída
3 4 1.2 0.0 0.0 0.0 0.0 -9.0 0.0 0.0 7.0 0.0 16.0 0.0	A matriz não é diagonal

Restrições

- 1) Seu programa deve definir uma função que não recebe argumentos, mas cria e retorna uma matriz lida da entrada padrão seguindo o formato especificado no enunciado.
- 2) Seu programa deve definir uma função que recebe como argumento uma matriz (pode receber como argumentos adicionais seu tamanho, mas é opcional) e retorna `True` ou `False`. O retorno será `True` caso a matriz seja diagonal e `False` caso contrário.
- 3) O corpo principal do programa deve utilizar as funções definidas nas Restrições 1 e 2, além de imprimir na saída padrão a conclusão conforme especificado no formato de saída.
- 4) Não é permitido o uso de bibliotecas como `numpy`, `pandas` ou similares.

Distribuição de Pontos

Esta questão vale 3,0 dos 10,0 pontos da avaliação. Os pontos são distribuídos em:

- Q2.1 - Definição da função de leitura, conforme descrito na Restrição 1: 0,5
- Q2.2 - Definição da função de verificação, conforme descrito na Restrição 2: 2,0
- Q2.3 - Corpo principal do programa, conforme descrito na Restrição 3: 0,5

QUESTÃO 3 - Soma de Vizinhos (em vermelho, errata apresentada durante a prova)

Escreva um programa que lê o tamanho da matriz A e, em seguida, seu conteúdo. Por fim, gere uma nova matriz B, onde cada elemento B_{ij} é a soma dos valores dos quatro vizinhos de A_{ij} (acima, abaixo, direita e esquerda) com o elemento A_{ij} .

Entrada

A entrada possui o mesmo formato que na Questão 2, de modo que a rotina de leitura implementada lá poderá ser reutilizada aqui. Para isso, basta chamá-la (não é preciso reescrever o código fonte).

Saída

O programa deve imprimir como saída a matriz B. O formato de saída é similar ao de entrada, ou seja, primeiro é impresso o tamanho da matriz e em seguida seu conteúdo, sendo os coeficientes de cada linha separados por um espaço em branco. A única diferença é que na impressão os valores devem ser apresentados com duas casas decimais.

Exemplo

Entrada	Saída
3 4 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 20.0 30.0	3 4 8.00 12.00 16.00 15.00 21.00 30.00 44.00 49.00 24.00 45.00 67.00 58.00

Restrições

- 1) Vide Restrição 1 da Questão 2, sendo que não é preciso reescrever o código.
- 2) Seu programa deve definir uma função que recebe como argumento uma matriz A (pode receber como argumentos adicionais seu tamanho, mas é opcional) e retorna uma matriz B, calculada conforme descrito no enunciado.
- 3) Seu programa deve definir um procedimento que recebe como argumento uma matriz (pode receber como argumentos adicionais seu tamanho, mas é opcional) e imprime na saída padrão seu conteúdo conforme a formatação especificada no enunciado.
- 4) O corpo principal do programa deve utilizar as funções definidas nas Restrições 1, 2 e 3.
- 5) Não é permitido o uso de bibliotecas como `numpy`, `pandas` ou similares.

Distribuição de Pontos

Esta questão vale 3,0 dos 10,0 pontos da avaliação. Os pontos são distribuídos em:

- Q3.1 - Definição da função de cálculo, conforme descrito na Restrição 2: 2,0
- Q3.2 - Definição do procedimento de escrita, conforme descrito na Restrição 3: 0,5
- Q3.3 - Corpo principal do programa, conforme descrito na Restrição 4: 0,5

QUESTÃO 4 - Remover ou Substituir

Escreva um programa que lê uma coleção de valores inteiros e, em seguida, lê um valor cujas ocorrências devem ser removidas da coleção original, um valor a ser buscado e outro que substituirá todas as ocorrências daquele que foi buscado.

Entrada

A primeira linha da entrada indica a quantidade N de valores que compõem a coleção original. As próximas N linhas contém os valores da coleção, um por linha. Em seguida é dada uma linha com o valor a ser removido da coleção original e, por fim, uma linha com o valor a ser buscado e o valor que o substituirá. Estes últimos são separados por um espaço em branco.

Saída

O programa deve imprimir o estado final da coleção de valores, colocando um valor por linha.

Exemplo

Entrada	Saída
5 10 30 10 20 50 10 50 40	30 20 40

Restrição

Não é permitida a criação de listas auxiliares.

Distribuição de Pontos

Esta questão vale 2,0 dos 10,0 pontos da avaliação. Os pontos são distribuídos em:

- Q4.1 - Leitura e escrita conforme especificado: 0,4
- Q4.2 - Remoção de valores, conforme especificado: 0,8
- Q4.3 - Substituição de valores, conforme especificado: 0,8