



Questão 3

Correto

Atingiu 2,00 de 2,00

Aprendendo Zip 2

Crie um programa que lê 5 números inteiros e armazena na lista *lista1* e depois lê mais 5 número inteiros e armazena na *lista2*.

Em seguida, crie e imprima a lista de tuplas *list_tuple* formada a partir dos elementos da *lista1* e *lista2*. (Dica: utilize a função zip.)

Por fim, imprima uma lista *medias* cujos elementos são as médias das tuplas.

Entrada

A entrada consiste em 10 números inteiros.

Saída

A saída consiste na lista de tuplas formada pelos elementos de *lista1* e *lista2*, e na lista das médias das tuplas.

Observações

- No primeiro exemplo de teste, foram digitados os elementos da *lista1*, 14, 23, 0, 4 e 5, e os elementos da *lista2*, 93, 84, 5, 43 e 1. Retornando as listas de tuplas [(14, 93), (23, 84), (0, 5), (4, 43), (5, 1)] e de médias de cada tupla [53.5, 53.5, 2.5, 23.5, 3.0].

For example:

Input	Result
14 23 0 4 5 93 84 5 43 1	[(14, 93), (23, 84), (0, 5), (4, 43), (5, 1)] [53.5, 53.5, 2.5, 23.5, 3.0]
0 0 0 0 0 0 0 0 0 0 0	[(0, 0), (0, 0), (0, 0), (0, 0), (0, 0)] [0.0, 0.0, 0.0, 0.0, 0.0]
1 2 3 4 5 6 7 8 9 10	[(1, 6), (2, 7), (3, 8), (4, 9), (5, 10)] [3.5, 4.5, 5.5, 6.5, 7.5]

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```
1 lista1 =[]
2 lista2 =[]
3 for i in range(10):
4     if i > 4:
```



```

5      lista2 +=[int(input())]
6  ▾    else:
7      lista1 +=[int(input())]
8  list_tuple = list(zip(lista1,lista2))
9  print(list_tuple)
10 list_media =[]
11 ▾ for i in range(5):
12     list_media += [(list_tuple[i][0]+list_tuple[i][1])/2]
13 print(list_media)
14

```

PRECHECK VERIFICAR



	Input	Expected	Got	
✓	14 23 0 4 5 93 84 5 43 1	[(14, 93), (23, 84), (0, 5), (4, 43), (5, 1)] [53.5, 53.5, 2.5, 23.5, 3.0]	[(14, 93), (23, 84), (0, 5), (4, 43), (5, 1)] [53.5, 53.5, 2.5, 23.5, 3.0]	✓
✓	0 0 0 0 0 0 0 0 0 0	[(0, 0), (0, 0), (0, 0), (0, 0), (0, 0)] [0.0, 0.0, 0.0, 0.0, 0.0]	[(0, 0), (0, 0), (0, 0), (0, 0), (0, 0)] [0.0, 0.0, 0.0, 0.0, 0.0]	✓
✓	1 2 3 4 5 6 7 8 9 10	[(1, 6), (2, 7), (3, 8), (4, 9), (5, 10)] [3.5, 4.5, 5.5, 6.5, 7.5]	[(1, 6), (2, 7), (3, 8), (4, 9), (5, 10)] [3.5, 4.5, 5.5, 6.5, 7.5]	✓
✓	-21 93 -4 5 1 8992 -2 1 43 56	[(-21, 8992), (93, -2), (-4, 1), (5, 43), (1, 56)] [4485.5, 45.5, -1.5, 24.0, 28.5]	[(-21, 8992), (93, -2), (-4, 1), (5, 43), (1, 56)] [4485.5, 45.5, -1.5, 24.0, 28.5]	✓



	Input	Expected	Got	
✓	4847395 37483 128983274 387492 192838 -849283 -283 2341 371 -9382	[(4847395, -849283), (37483, -283), (128983274, 2341), (387492, 371), (192838, -9382)] [1999056.0, 18600.0, 64492807.5, 193931.5, 91728.0]	[(4847395, -849283), (37483, -283), (128983274, 2341), (387492, 371), (192838, -9382)] [1999056.0, 18600.0, 64492807.5, 193931.5, 91728.0]	✓
✓	-1 1 -2 2 -3 3 -4 4 -5 5	[(-1, 3), (1, -4), (-2, 4), (2, -5), (-3, 5)] [1.0, -1.5, 1.0, -1.5, 1.0]	[(-1, 3), (1, -4), (-2, 4), (2, -5), (-3, 5)] [1.0, -1.5, 1.0, -1.5, 1.0]	✓

Passou em todos os teste! ✓

Correto

Notas para este envio: 2,00/2,00.

