

Dicas para o Moodle

Informações de submissão e verificação:

https://ava.ufabc.edu.br/pluginfile.php/24501/mod_resource/content/0/PI_Submissao_EP_2020Q1.pdf

https://ava.ufabc.edu.br/pluginfile.php/24502/mod_resource/content/0/PI_Submissao_EP_2020Q1_Python.pdf

Notem que a nota apresentada pelo Moodle não é a nota final, mas apenas um indicativo. Os docentes da disciplina avaliarão individualmente a corretude do programa e questões de plágio!

A estrutura básica de um exercício do Moodle geralmente é:

```
def funcao(x):  
    instruções  
    return resultado
```

```
x = tipo(input())  
resultado = funcao(x)  
print(resultado)
```

A `funcao` é onde você deve escrever o algoritmo para resolução do problema. Em alguns casos pode ser necessário mais do que uma função.

Dentro da função temos uma ou mais instruções que computam o resultado sempre terminando com um `return resultado`, com `resultado` sendo a variável que armazena a resposta.

O bloco de instruções seguintes, que não fazem parte da função, é chamado de **bloco principal** e contém as instruções de captura de entrada e impressão da saída.

Para capturar uma entrada utilizamos a função `input` que **sempre** retorna uma *string*. Caso o tipo requisitado seja diferente, devemos também chamar a função de conversão. Exemplos:

```
x = int(input()) # captura um inteiro  
y = float(input()) # captura um número real  
s = input() # captura uma string
```

A saída deve obedecer **exatamente** como instruído no enunciado do problema! Exemplos:

- No EP1.1 pede como saída a string “Alo, mundo!”, o único `print` deve ser `print("Alo, mundo!")`. Exemplo de **saídas erradas**:

```
print("Alô, mundo!")  
print("Alo mundo!")  
print("Alo, mundo! ")
```

- No EP1.2 pede como saída a média das notas com **duas** casas decimais, nesse caso deve-se utilizar a função `round` com o número de casas decimais no segundo parâmetro:

```
media = round(calculaMedia(x,y,z), 2)
print(f"media: .2f")
```

Reparem que para esse caso utilizamos uma **string de formatação** que permite controlar o número de casas decimais: `f"{variavel:.nf}"`, variável é o nome da variável e `n` é a quantidade de casas decimais desejada. Isso se faz necessário pois se, por exemplo, o resultado for `4.0` a função `round` descartará valores 0 a direita após o ponto decimal.

Note que em alguns exercícios é pedido um número diferente de casas decimais. A função `print` não imprime mais nada, apenas o resultado!

Exemplos de **saídas erradas**:

```
print(f"A media é: {media}")
print("A media é: ")
print(media)
```

A entrada para esse exercício devem ser três valores reais, ou seja, usamos `x = float(input())`. Exemplo de **entradas erradas**:

```
x = input()
x = float(input("Entre com x: "))
float(x) = input()
```

Quando o exercício especificar que as entradas serão em uma determinada ordem, os comandos `input` devem ser feitos **nessa ordem**. Exemplo, no EP1.4 ele especifica que a entrada será `Ax`, `Ay`, `Bx`, `By` nessa ordem, portanto deve ser feito:

```
Ax = float(input())
Ay = float(input())
Bx = float(input())
By = float(input())
```

Exemplo de **entradas erradas**:

```
Ax = float(input())
Bx = float(input())
Ay = float(input())
By = float(input())

Ax = input()
Ay = input()
Bx = input()
By = input()
```

```
Ax = float(input("Ax"))
Ay = float(input("Ay"))
Bx = float(input("Bx"))
By = float(input("By"))
```

No EP1.8 é pedido **quatro** saídas diferentes. Para esses casos você pode utilizar a função `print` dentro da função. Cada saída deverá ser um `print`. Exemplo:

```
def funcao(x):
    instrucoes
    print(resposta1)
    print(resposta2)
    print(resposta3)
    print(resposta4)
```

```
x = int(input())
funcao(x)
```

Note que não devemos utilizar `print(funcao(x))` nesse caso!