

1. Quais estruturas são obrigatórias na criação de uma função?
  - a) if, else, return, def, comando da função
  - b) def, return, comando da função
  - c) def, comando da função**
  - d) break, def, return
  - e) print, def
2. Para implementar um algoritmo que calcula o IMC, foi utilizada uma função "calcula\_IMC". Qual das seguintes implementações foi feita da forma correta?

a)

```
p = float(input("Digite o seu peso: "))
h = float(input("Digite a sua altura: "))

def calcula_IMC(p,h):
    IMC = p/h**2
    print("Seu IMC é de ", IMC)

calcula_IMC(p,h)
```

b)

```
p = float(input("Digite o seu peso: "))
h = float(input("Digite a sua altura: "))

def calcula_IMC(p,h)
    IMC = p/h**2
    print("Seu IMC é de ", IMC)

calcula_IMC(p,h)
```

c)

```
p = float(input("Digite o seu peso: "))
h = float(input("Digite a sua altura: "))

def calcula_IMC():
    IMC = p/h**2
    print("Seu IMC é de ", IMC)

calcula_IMC(p,h)
```

```
p = float(input("Digite o seu peso: "))
h = float(input("Digite a sua altura: "))

def calcula_IMC(p,h):
    IMC = p/h**2
    print("Seu IMC é de ", IMC)

calcula_IMC()
```

d)

```
p = float(input("Digite o seu peso: "))
h = float(input("Digite a sua altura: "))

def calcula_IMC():
    IMC = p/h**2
    print("Seu IMC é de ", IMC)

calcula_IMC()
```

e)

3. O que é um objeto de uma classe?

- a) É um conjunto de entidades
- b) É um elemento computacional que representa alguma entidade**
- c) É um bloco de comandos
- d) É um vetor
- e) É uma biblioteca

4. Qual dos algoritmos abaixo está implementando a classe “carnes” corretamente?

```
class carnes:

    def __init__(self, nome, precokg):
        self.nome = nome
        self.precokg = precokg

carne1 = carnes("Alcatra", 25)
```

a)

```
class carnes:

    def __init__(self, nome, precokg):
        self.nome = nome
        self.precokg = precokg

carne1 = carnes("Alcatra", 25)
```

b)

```
class carnes:

    def __init__():
        self.nome = nome
        self.precokg = precokg

carne1 = carnes("Alcatra", 25)
```

c)

```
class carnes:

def __init__(self, nome, precokg):
    self.nome = nome
    self.precokg = precokg

carne1 = carnes("Alcatra", 25)
```

d)

```
class carnes:

    def __init__(self, nome, precokg):
        self.nome = nome
        self.precokg = precokg

carne1 = carnes("Alcatra", 25)
```

e)