Nome(s): \_\_\_Lucas\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**As respostas devem ser um print do código e da saída em python**

**1) Vamos criar um conjunto:**

A = {1,2,3,4,5,6}

****print(A)

**Saída:**

**2) Vamos criar um conjunto a partir de uma lista**

lista = [“bananas”, “peras”, “laranjas”, “abacates”]

B = set(lista)

print(B)

**Saída:**

**3) Seguindo a mesma lógica do item anterior:**

lista = [“bananas”, “peras”, “laranjas”, “limões”, “bananas”, “bananas”, “abacates”, “laranjas”]

B = set(lista)

print(B)

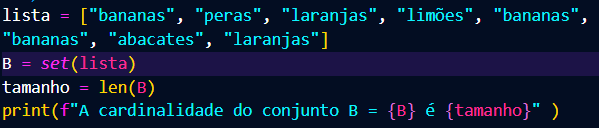
**Saída:**

**Comparando os itens 2 e 3, a que conclusão podemos chegar?**

**Resposta: Conjunto não repete elemento.**

**4) Imprima a cardinalidade do conjunto B obtido no item 3 da forma: “A cardinalidade do conjunto B = { ... } é {tamanho}”**

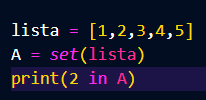
**Dica: utilize a palavra reservada do python “len”**

**Resposta:**

**5) Teste as relações de pertinência e imprima a resposta (A = {1,2,3,4,5})**

**Dica: utilize a palavra reservada do python “in”**

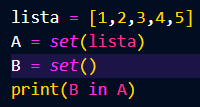
a)



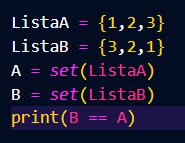
b)



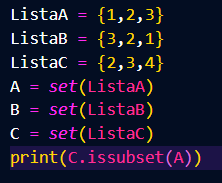
c)



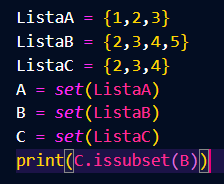
**6) Teste a igualdade entre os conjuntos A= {1,2,3} e B = {3,2,1}, A é igual a B? Imprima o resultado**

****

****

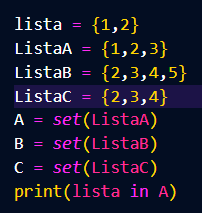
**7) Utilize a função issubset() para testar todos os subconjuntos de C = {2,3,4} – imprima os resultados**

****

****

****

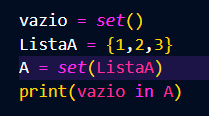
**Agora, faça o teste utilizando o operador de pertinência em python para o seguinte exemplo:**

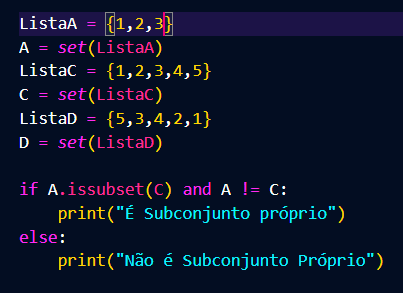
**Qual resultado é esperado? O python respeita esse resultado?**

**R: Resultado falso, Python entende que conjuntos não pertencem a conjuntos.**

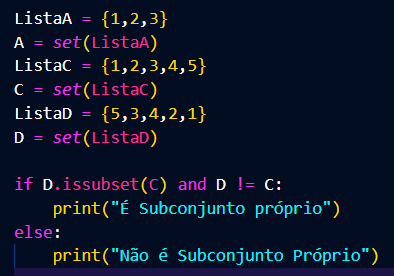
**Faça o teste para o conjunto vazio:**

**Qual resultado é esperado? O python respeita esse resultado?**

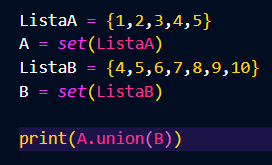
** R: Falso também, conjunto vazio ainda é um conjunto. Python dá como falso.**

**8) Crie uma verificação para testar se A = {1,2,3} é subconjunto próprio de C = {1,2,3,4,5} – imprima o código e resultado. Agora reaproveite o código para testar se D é subconjunto próprio de C.**

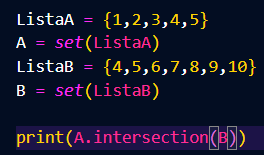
****

****

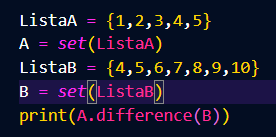
****

**9) Considerando: A = {1,2,3,4,5} e B = {4,5,6,7,8,9,10} faça a conta (mostrando a simbologia matemática e imprima os resultados em python):  
a)**

****

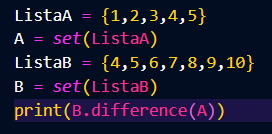
**b)**

****

**c) A – B**

****

**d) B – A**

****

**10) Faça um menu que só encerre quando o usuário solicitar (opção de sair) que seja interativo e com as devidas validações de possíveis erros de entrada do usuário. O objetivo é fazer a operação entre 2 conjuntos, ou seja, crie uma forma de pedir dois conjuntos para o usuário (conjuntos A e B – posteriormente esses conjuntos podem ser alterados pelo usuário). As opções de operações são:**

**a) União**

**b) Intersecção**

**c) Diferença**

**d) Produto cartesiano**

**d) Verificação se A é subconjunto de B (submenu: subconjunto ou subconjunto próprio)**

**e) Mesma verificação do item e, mas de B com A.**