Nome(s: Lucas Stopinski da Silva

**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:** **Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**

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

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

B = set(lista)

print(B)

**Saída:** **Interface gráfica do usuário, Texto

Descrição gerada automaticamente**

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

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

B = set(lista)

print(B)

**Saída:** **Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**

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

**Resposta:** O set() faz com que os elementos duplicados aparecem apenas uma vez.

**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:** **Tela de celular com aplicativo aberto

Descrição gerada automaticamente**

**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) Texto

Descrição gerada automaticamente

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

**Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**

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

**Texto

Descrição gerada automaticamente**

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

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

**Não respeita, pois E é para elementos aqui estamos falando de um conjunto**

**Texto

Descrição gerada automaticamente**

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

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

**Respeita sim, todo conjunto vazio faz parte do conjunto e o símbolo está correto.**

**Texto

Descrição gerada automaticamente**

**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 = {5,3,4,2,1} é subconjunto próprio de C.**

**Texto

Descrição gerada automaticamente**

**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)**

**Texto

Descrição gerada automaticamente**

**b)**

**Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**

**c) A – B**

**Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**

**d) B – A**

**Texto

Descrição gerada automaticamente**

**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.**

def obter\_conjunto(nome):

    while True:

        try:

            conjunto = input(f"📝 Digite os elementos do conjunto {nome} separados por vírgula: ")

            return set(map(int, conjunto.split(',')))

        except ValueError:

            print("❌ Entrada inválida! Certifique-se de digitar números inteiros separados por vírgula.")

def uniao(A, B):

    print("🌐 A ∪ B =", A | B)

def intersecao(A, B):

    print("🔗 A ∩ B =", A & B)

def diferenca(A, B):

    print("➖ A - B =", A - B)

    print("➖ B - A =", B - A)

def produto\_cartesiano(A, B):

    produto\_cartesiano = {(a, b) for a in A for b in B}

    print("✖️ A × B =", produto\_cartesiano)

def verificar\_subconjunto\_A\_B(A, B):

    if A.issubset(B):

        print("📑 A é subconjunto de B")

        if A != B:

            print("📄 A é subconjunto próprio de B")

    else:

        print("❌ A não é subconjunto de B")

def verificar\_subconjunto\_B\_A(B, A):

    if B.issubset(A):

        print("📑 B é subconjunto de A")

        if B != A:

            print("📄 B é subconjunto próprio de A")

    else:

        print("❌ B não é subconjunto de A")

def alterar\_conjuntos():

*global* A, B

    A = obter\_conjunto('A')

    B = obter\_conjunto('B')

def sair():

    print("🚪 Encerrando o programa...")

    return True

def menu\_principal():

    switch = {

        '1': lambda: uniao(A, B),

        '2': lambda: intersecao(A, B),

        '3': lambda: diferenca(A, B),

        '4': lambda: produto\_cartesiano(A, B),

        '5': lambda: verificar\_subconjunto\_A\_B(A, B),

        '6': lambda: verificar\_subconjunto\_B\_A(B, A),

        '7': alterar\_conjuntos,

        '0': sair

    }

    while True:

        print("\n===========================================")

        print("            📊 MENU DE OPERAÇÕES 📊         ")

        print("===========================================")

        print("🔢 Escolha uma das opções abaixo:")

        print(" 1️⃣  União (A ∪ B) 🌐")

        print(" 2️⃣  Intersecção (A ∩ B) 🔗")

        print(" 3️⃣  Diferença (A - B ou B - A) ➖")

        print(" 4️⃣  Produto Cartesiano (A × B) ✖️")

        print(" 5️⃣  Verificar se A é subconjunto de B 📑")

        print(" 6️⃣  Verificar se B é subconjunto de A 📄")

        print(" 7️⃣  Alterar conjuntos A e B 🔄")

        print(" 0️⃣  Sair do programa 🚪")

        print("===========================================")

        opcao = input("👉 Digite a sua escolha: ")

        if opcao in switch:

            if opcao == '0' and switch[opcao]():

                break

            else:

                switch[opcao]()

        else:

            print("❌ Opção inválida! Por favor, escolha uma opção válida.")

if \_\_name\_\_ == "\_\_main\_\_":

    A = obter\_conjunto('A')

    B = obter\_conjunto('B')

    menu\_principal()