1. Em uma cidade, quatro estações meteorológicas realizam o registro da temperatura durante cinco dias consecutivos. Desenvolva um programa em Python que auxilie na análise desses dados.

**Requisitos do Exercício:**

1. **Matriz de Dados:**
   * Crie uma matriz (lista de listas) para armazenar as temperaturas.
   * Cada linha da matriz representará uma estação meteorológica, e cada coluna representará um dia.
2. **Entrada de Dados e Repetições:**
   * Utilize repetições (loops) para solicitar ao usuário que informe, para cada estação e para cada dia, o valor da temperatura registrada.
   * Certifique-se de armazenar os dados na matriz de forma organizada.
3. **Cálculos com Repetições:**
   * Utilize loops para percorrer a matriz e calcular a média de temperatura de cada estação.
   * Calcule também a média geral das temperaturas registradas em todas as estações durante os cinco dias.
4. **Condições:**
   * Implemente estruturas condicionais para verificar se alguma estação registrou temperaturas acima de 30°C em pelo menos três dos cinco dias.
   * Caso essa condição seja satisfeita para alguma estação, exiba uma mensagem de alerta informando que a estação apresentou temperaturas elevadas.
5. **Funcionalidades Adicionais:**
   * Identifique e exiba o dia em que cada estação registrou a maior temperatura. Adicione isto em um dicionário no formato: “estacao1”:X, “estacao2”:Y, etc.
   * Verifique qual estação teve a maior média de temperatura e apresente essa informação.

2. Criar um programa em Python que simule um sistema de gerenciamento de biblioteca. O sistema deve utilizar dicionários para armazenar os livros, seus autores e status de empréstimo, além de implementar funcionalidades de cadastro, consulta, empréstimo e devolução de livros.

**Funcionalidades:**

1. **Cadastro de Livros:**
   * O sistema deve permitir cadastrar novos livros, informando título, autor e status de empréstimo (disponível ou emprestado).
   * Os dados dos livros devem ser armazenados em um dicionário, onde a chave é o título do livro e o valor é outro dicionário contendo o autor e o status.
2. **Consulta de Livros:**
   * O usuário deve poder consultar as informações de um livro específico, informando o título do livro.
   * O sistema deve exibir o título, autor e status do livro consultado.
3. **Empréstimo de Livros:**
   * O usuário deve poder realizar o empréstimo de um livro, informando o título do livro.
   * O sistema deve verificar se o livro está disponível e, em caso afirmativo, atualizar o status do livro para "emprestado".
   * Caso o livro não esteja disponível, o sistema deve exibir uma mensagem informando que o livro já está emprestado.
4. **Devolução de Livros:**
   * O usuário deve poder realizar a devolução de um livro, informando o título do livro.
   * O sistema deve atualizar o status do livro para "disponível".

**Estrutura de Dados:**

Utilize um dicionário para armazenar os dados dos livros. A chave do dicionário será o título do livro, e o valor será outro dicionário contendo as seguintes informações:

* "autor": Nome do autor do livro (string)
* "status": Status do empréstimo do livro (string, "disponível" ou "emprestado")

**Exemplo de Dicionário:**

Python

livros = {

"Dom Quixote": {"autor": "Miguel de Cervantes", "status": "disponível"},

"1984": {"autor": "George Orwell", "status": "emprestado"},

"O Senhor dos Anéis": {"autor": "J.R.R. Tolkien", "status": "disponível"},

}

**Implementação:**

1. Crie uma função para cadastrar novos livros no dicionário de livros.
2. Crie uma função para consultar as informações de um livro específico.
3. Crie uma função para realizar o empréstimo de um livro.
4. Crie uma função para realizar a devolução de um livro.
5. Crie um menu principal que permita ao usuário escolher entre as funcionalidades do sistema.