

Lista de Exercícios sobre Ownership

1. Clonando e Movendo Strings

Escreva um programa que recebe duas strings como entrada. A primeira string deve ser movida para uma nova variável e a segunda string deve ser clonada para uma nova variável. Mostre os valores das strings originais e das novas após o movimento e clonagem. Explique a diferença entre mover e clonar.

2. Função de Soma com Empréstimo Imutável

Crie uma função soma que recebe uma slice de números inteiros como uma referência imutável. A função deve calcular e retornar a soma dos números. No programa principal, crie um vetor de inteiros e passe uma slice para a função. Após chamar a função, mostre o valor do vetor original. Dica: procure por `vec!`.

3. Reatribuição e Ownership

Crie um programa que tenha duas variáveis que armazenam strings. Reatribua uma das strings para a outra variável e mostre o que acontece com o valor da variável original após a reatribuição.

4. Empréstimo Mutável

Escreva uma função `adicionar_prefixo` que recebe uma string mutável e adiciona o prefixo "Prefixo: " a essa string. No programa principal, crie uma string e passe-a para a função como referência mutável. Mostre o valor da string após a modificação.

5. Multiplicação de Vetores com Empréstimo Imutável

Crie uma função que recebe dois vetores de números inteiros como referências imutáveis e retorna um novo vetor contendo os produtos dos elementos correspondentes. O programa principal deve criar dois vetores, passar ambos para a função, e depois imprimir o vetor resultante. Dica: procure por `Vec<>` e `vec!`

6. Número de Caracteres com Empréstimo Imutável

Implemente uma função chamada `contar_caracteres` que recebe uma referência imutável para uma string e retorna o número de caracteres nela. No programa principal, crie uma string e chame a função, mostrando o resultado. Dica: procure por `chars` e `count`.

7. Split de String com Empréstimo Imutável

Escreva uma função que recebe uma referência imutável para uma string e retorna um vetor de palavras que são divididas por espaço. No programa principal, crie uma string, passe-a para a função e mostre o vetor resultante. Dica: procure por `split_whitespace` e `collect`.

8. Função com Várias Referências Imutáveis

Escreva uma função concatenar que recebe duas referências imutáveis para strings e retorna uma nova string que é a concatenação das duas. No programa principal, passe duas strings para a função e mostre o resultado.

9. Função com Referência Mutável e Imutável

Implemente uma função adicionar_elemento que recebe uma referência mutável para um vetor de inteiros e uma referência imutável para um número inteiro. A função deve adicionar o número ao vetor. No programa principal, crie um vetor e um número, e depois passe-os para a função. Dica: procure por Vec<>

10. Ciclo de Empréstimo Mutável

Crie um programa que recebe um vetor de números inteiros. O programa deve usar uma função que altera o vetor dobrando o valor de cada elemento. Use referências mutáveis para modificar o vetor, sem perder o controle do *ownership*. Mostre o vetor após a modificação. Dica: procure por Vec<>