

Exercícios de HashMaps e Erros

1. Mapeamento de Parentesco

Crie uma estrutura que modele um sistema de árvore genealógica. Utilize um `HashMap<String, Vec<String>>` para representar a relação entre pais e filhos, onde cada chave é o nome de uma pessoa e o valor é uma lista dos nomes de seus filhos. Implemente uma função que, dada uma pessoa, retorne todos os seus descendentes diretos e indiretos.

2. Contagem de Caracteres Repetidos com Unicode

Modifique o exercício de contagem de ocorrências de palavras para contar a frequência de caracteres em uma string que contém caracteres Unicode. Utilize um `HashMap<char, u32>` para armazenar as frequências.

3. Colapso de HashMap com Chaves Compostas

Dado um `HashMap<(String, String), u32>`, onde a primeira parte da chave é uma categoria e a segunda é um item, crie uma função que retorne um novo `HashMap<String, u32>` colapsando todas as categorias e somando os valores de seus itens correspondentes.

4. Diferença entre HashMaps

Implemente uma função que receba dois `HashMap<String, u32>` e retorne um novo `HashMap<String, u32>` contendo apenas os pares chave-valor presentes no primeiro `HashMap`, mas não no segundo.

5. HashMap com Tuplas como Valores

Crie um sistema de inventário para um jogo, onde cada item é mapeado para uma tupla `(quantidade, valor)`. O programa deve permitir que o jogador adicione itens, remova itens e calcule o valor total do inventário.

6. Cálculo de Média com Erros em Entradas

Implemente uma função que receba um vetor de strings representando números e calcule a média apenas dos números válidos. A função deve retornar um `Result<f64, String>` que contenha a média ou uma mensagem de erro se nenhum número for válido.

7. Mapeamento de Resultados de Funções

Dado um vetor de funções que retornam um `Result<i32, String>`, implemente uma função que execute todas as funções em sequência e colete todos os resultados bem-sucedidos em um vetor. Se todas as funções falharem, retorne um `Err` com uma mensagem de erro agregada.

8. Processamento de Arquivos em Lote com Erro Detalhado

Escreva uma função que recebe uma lista de caminhos para arquivos, lê o conteúdo de cada um e retorna um ``Result<HashMap<String, String>, Vec<String>>``. A chave do ``HashMap`` deve ser o nome do arquivo e o valor seu conteúdo. Em caso de falha, o retorno deve conter uma lista dos arquivos que não puderam ser lidos.

9. Conversão Segura com Erros Detalhados

Implemente uma função que receba um vetor de strings e tente convertê-las para números inteiros. Para cada conversão falha, adicione um detalhe do erro (como o índice da string falha) em um vetor de erros. A função deve retornar um ``Result<Vec<i32>, Vec<String>>``.

10. Erro ao Aplicar Funções a HashMaps

Escreva uma função que aplique uma operação binária entre os valores de dois ``HashMap<String, i32>``. Se qualquer chave estiver presente em apenas um dos mapas, a função deve retornar um erro detalhado informando quais chaves estavam ausentes. Retorne um ``Result<HashMap<String, i32>, Vec<String>>``.