

prg101 – Crie uma função `soma_numeros(a, b, c)` que recebe três números como parâmetros e retorna a soma deles. Peça ao usuário para inserir os valores e exiba o resultado.

prg102 – Crie uma função `eh_palindromo(texto)` que recebe uma string e retorna `True` se for um palíndromo (ex: "arara") ou `False` caso contrário.

prg103 – Crie uma função `maior_menor(lista)` que recebe uma lista de números e retorna o maior e o menor valor (em uma tupla). Exemplo de entrada: [5, 2, 9, 1] → Saída: (9, 1).

prg104 – Crie uma função `inverter_lista(lista)` que recebe uma lista e retorna uma nova lista com os elementos invertidos (sem usar `[::-1]` ou métodos prontos).

prg105 – Use `random.randint(1, 100)` para gerar um número aleatório. O usuário deve adivinhar o número, e o programa informa se o chute foi alto, baixo ou correto (use funções).

prg106 – Crie uma função `sorteio_nomes(lista_nomes, n)` que recebe uma lista de nomes e um número `n`, e retorna `n` nomes aleatórios sem repetição (use `random.randint`).

prg107 – Crie uma função `contar_linhas(arquivo.txt)` que lê um arquivo `.txt` e retorna o número de linhas. Teste com um arquivo criado por você.

prg108 – Crie um programa que gera 10 números aleatórios (entre 1 e 100) usando `randint` e os salva em um arquivo `numeros.txt` (um por linha). Depois, leia o arquivo e exiba a soma dos números.

prg109 – Use um arquivo `palavras.txt` com uma palavra por linha. Leia o arquivo e escolha uma palavra aleatória. Crie uma função `jogar_forca(palavra)` que gerencia o jogo (chances limitadas, exibe letras acertadas).

Dica: Use listas para controlar letras já tentadas.