Lista de Exercícios 01
Professor: Mauricio de Souza Estevam

1. Escreva uma função soma(a, b) que receba dois números e retorne a soma deles.
2. Crie uma função maior(a, b) que receba dois números e retorne o maior entre eles.
3. Escreva uma função eh_par(numero) que receba um número e retorne True se ele for par e False caso contrário.
4. Implemente uma função quadrado(x) que receba um número e retorne o seu quadrado.
5. Crie uma função mult_tres_numeros(a, b, c) que multiplica três números e retorna o resultado.
6. Escreva uma função soma_lista(lista) que receba uma lista de números e retorne a soma de todos os elementos.
7. Crie uma função conta_vogais(texto) que retorne o número de vogais em uma string.
8. Implemente uma função inverte_string(s) que retorne a string s invertida.
9. Escreva uma função fatorial(n) que calcule o fatorial de um número inteiro n.
10. Crie uma função eh_primo(n) que verifique se um número é primo.
11. Crie uma função conta_palavras(texto) que conte quantas palavras existem em uma string.
12. Escreva uma função eh_palindromo(s) que verifique se uma palavra ou frase é um palíndromo.
13. Implemente uma função conta_letras(s, letra) que conte quantas vezes uma determinada letra aparece em uma string.
14. Crie uma função capitaliza_palavras(texto) que capitaliza a primeira letra de cada palavra em um texto.
15. Escreva uma função ocorrencias_palavras(texto) que retorne um dicionário com a contagem de cada palavra em um texto.
16. Crie uma função retira_espacos(texto) que retorne uma string sem espaços em branco.
17. Implemente uma função encontra_palavra(texto, palavra) que retorne o índice da primeira ocorrência de uma palavra em uma string.
18. Escreva uma função alterna_maiusculas(texto) que alterne as letras para maiúsculas e minúsculas.
19. Crie uma função apaga_vogais(s) que remova todas as vogais de uma string.
20. Implemente uma função troca_vogais(s, nova_letra) que substitua todas as vogais de uma string por uma nova letra.
21. Crie uma função media_lista(lista) que receba uma lista de números e retorne a média.

22. Escreva uma função maior_elemento(lista) que retorne o maior número de uma lista.
23. Implemente uma função menor_elemento(lista) que retorne o menor número de uma lista.
24. Crie uma função conta_ocorrencias(lista, elemento) que conta quantas vezes um elemento aparece em uma lista.
25. Escreva uma função remove_duplicados(lista) que retorne uma lista sem elementos duplicados.
26. Crie uma função soma_pares(lista) que retorne a soma de todos os números pares em uma lista.
27. Implemente uma função elementos_unicos(lista) que retorne uma lista com os elementos únicos.
28. Escreva uma função lista_invertida(lista) que retorne a lista invertida.
29. Crie uma função intersecao_listas(lista1, lista2) que retorne os elementos em comum entre duas listas.
30. Implemente uma função uniao_listas(lista1, lista2) que retorne a união de duas listas.
31. Escreva uma função potencia(base, expoente) que calcule base elevado a expoente.
32. Crie uma função raiz_quadrada(n) que retorne a raiz quadrada de um número.
33. Implemente uma função area_circulo(raio) que calcule a área de um círculo de raio r.
34. Escreva uma função hipotenusa(a, b) que calcule a hipotenusa de um triângulo retângulo.
35. Crie uma função area_triangulo(base, altura) que calcule a área de um triângulo.
36. Implemente uma função eh_perfeito(n) que verifica se um número é um número perfeito.
37. Escreva uma função eh_armstrong(n) que verifica se um número é um número de Armstrong.
38. Crie uma função média_harmonica(lista) que calcula a média harmônica de uma lista de números.
39. Implemente uma função aproxima_pi(n) que calcula uma aproximação do número pi usando n termos da série de Leibniz.
40. Escreva uma função mediana(lista) que retorne a mediana de uma lista de números.
41. Implemente a função fibonacci(n) que retorna o n-ésimo termo da sequência de Fibonacci.
42. Crie uma função soma_recursiva(n) que soma todos os números de 1 até n recursivamente.
43. Escreva uma função recursiva produto(a, b) que multiplica dois números inteiros a e b.

44. Implemente uma função potencia_recursiva(base, exp) que calcula a potência de forma recursiva.
45. Crie uma função somatorio_lista_recursivo(lista) que retorne a soma dos elementos de uma lista recursivamente.
46. Escreva uma função inverte_string_recursiva(s) que inverta uma string recursivamente.
47. Implemente uma função conta_ocorrencias_recursiva(lista, elem) que conte o número de ocorrências de um elemento em uma lista de forma recursiva.
48. Crie uma função mcd(a, b) que calcule o máximo divisor comum de dois números usando recursão.
49. Escreva uma função mdc_lista(lista) que retorne o maior divisor comum entre os números de uma lista.
50. Implemente a função torre_hanoi(n, origem, destino, auxiliar) para resolver o problema da Torre de Hanói.
51. Crie uma função verifica_ano_bissexto(ano) que retorna True se o ano for bissexto.
52. Escreva uma função soma_diagonais(matriz) que retorne a soma dos elementos das diagonais de uma matriz.
53. Implemente uma função conta_ocorrencias_caracteres(s) que retorne um dicionário com a contagem de cada caractere em uma string.
54. Crie uma função media_notas(dicionario) que recebe um dicionário de alunos e notas e retorna a média das notas.
55. Escreva uma função converte_segundos(h, m, s) que converte horas, minutos e segundos em segundos.
56. Crie uma função gera_fibonacci_lista(n) que gera uma lista com os n primeiros números de Fibonacci.
57. Implemente uma função substitui_espaco(texto, simbolo) que substitua todos os espaços em uma string por um símbolo específico.
58. Escreva uma função conta_maiusculas(texto) que conta o número de letras maiúsculas em um texto.
59. Crie uma função ordena_lista(lista) que retorne a lista ordenada em ordem crescente.
60. Implemente uma função filtra_pares(lista) que retorne uma lista apenas com os números pares.
61. Crie uma função bubblesort(lista) que ordene uma lista usando o algoritmo Bubble Sort.
62. Implemente uma função valida_senha(senha) que verifique se uma senha atende a requisitos (tamanho, caracteres especiais, etc.).
63. Escreva uma função calcula_juros_compostos(capital, taxa, tempo) que retorne o montante final.

64. Crie uma função gera_senha(tamanho) que gere uma senha aleatória de um tamanho específico.
65. Implemente uma função dec2bin(n) que converte um número decimal para binário.
66. Escreva uma função bin2dec(binario) que converte um número binário para decimal.
67. Crie uma função valida_email(email) que verifique se um e-mail é válido.
68. Implemente uma função tamanho_arquivo(nome_arquivo) que retorne o tamanho de um arquivo em bytes.
69. Escreva uma função minimax(lista) que retorne o menor e o maior número de uma lista.
70. Crie uma função classifica_triangulo(a, b, c) que classifique um triângulo com base em seus lados.
71. Crie uma função conta_elementos(lista) que retorne um dicionário com a contagem de cada elemento na lista.
72. Escreva uma função merge_dicionarios(d1, d2) que una dois dicionários, somando os valores das chaves iguais.
73. Implemente uma função filtro_dicionario(dic, valor) que retorne um novo dicionário apenas com as chaves onde os valores são maiores que valor.
74. Crie uma função ordena_por_valores(dic) que ordene um dicionário por seus valores e retorne o resultado.
75. Escreva uma função inverte_dicionario(dic) que inverta as chaves e valores de um dicionário.
76. Implemente uma função escreve_arquivo(nome_arquivo, texto) que crie um arquivo com um texto.
77. Crie uma função le_arquivo(nome_arquivo) que leia o conteúdo de um arquivo e retorne uma string.
78. Escreva uma função conta_linhas(nome_arquivo) que retorne o número de linhas de um arquivo.
79. Implemente uma função conta_palavras_arquivo(nome_arquivo) que retorne o número total de palavras em um arquivo.
80. Crie uma função substitui_texto_arquivo(nome_arquivo, palavra_antiga, palavra_nova) que substitua todas as ocorrências de uma palavra em um arquivo por outra.
81. Crie uma função data_atual() que retorne a data atual no formato "dd/mm/aaaa".
82. Escreva uma função dias_entre_datas(data1, data2) que retorne a diferença em dias entre duas datas.
83. Implemente uma função adiciona_dias(data, n) que adicione n dias a uma data e retorne o novo valor.

84. Crie uma função formato_12h_para_24h(hora_12h) que converta uma hora no formato 12h para 24h.
85. Escreva uma função formato_24h_para_12h(hora_24h) que converta uma hora no formato 24h para 12h.
86. Crie uma função geradora conta_ate_n(n) que gere números de 1 até n.
87. Implemente um gerador pares(n) que gere todos os números pares de 1 até n.
88. Escreva um gerador quadrados(n) que gere os quadrados dos números de 1 até n.
89. Crie um gerador fibonacci_gen(n) que gere os primeiros n termos da sequência de Fibonacci.

.

May the force be whit you