

Listas

1. Faça um programa que recebe uma lista com o nome de 5 modelos de carros, outra com os preços e outra com o consumo de cada modelo. Mostre a média de preços dos modelos e o modelo com consumo mais econômico. Além disso, para os modelos que tem preço maior que a média de preços, mostre quantos litros são necessários para percorrer uma distância de 1000 Km.
2. Defina uma função `posicoes(lista, valor)` que tem como parâmetros uma lista de inteiros e um inteiro. A função deve retornar uma lista com todas as posições em que o valor aparece na lista. Construa um programa que recebe a quantidade de elementos de uma lista de números e seus elementos. Seu programa deve mostrar o menor elemento da lista e todas as posições em que ele aparece.
3. Uma empresa com 10 empregados quer dar aumento de acordo com a quantidade de anos trabalhados na empresa, do salário atual e da média salarial da empresa. Os trabalhadores que vão ter aumento são aqueles que tem tempo de serviço maior que 5 anos ou salário inferior ao valor da média salarial. Funcionários que têm as duas condições satisfeitas ganham aumento de 35%. Apenas a condição do tempo ganha 10% de aumento e apenas a condição do salário ganha aumento de 10%. Faça um programa que atualiza os salários e mostra o nome e o novo salário de cada funcionário que teve aumento.
4. Defina uma função `intersecao(lista1, lista2)` que tem como parâmetros duas listas. A função deve retornar uma lista com os elementos que aparecem nas duas listas. Faça um programa em que o usuário digita a quantidade de funcionários de uma empresa e seus nomes. Além disso, o usuário também deve digitar a quantidade de gerentes e os seus nomes. O programa deve mostrar os nomes dos funcionários que são gerentes.
5. Apresente um programa que recebe os 20 elementos de uma lista de números. Em seguida, seu programa deve trocar o primeiro elemento com o último, o segundo elemento com o penúltimo, etc., até trocar o 10º com o 11º. Mostre lista modificada.
6. Elabore um programa que recebe os elementos de uma lista de tamanho definido pelo usuário. Depois seu programa deve receber um número digitado pelo usuário e apagar da lista todas as ocorrências desse valor.
7. Faça um programa que recebe os elementos de uma lista com 10 inteiros. O programa deve mostrar os números primos e suas respectivas posições. Você pode criar uma função `primo(n)` que recebe um número como parâmetro e retorna `True` se `n` é primo e retorna `False` se `n` não é primo.
8. Defina uma função `media_lista(lista)` que tem como parâmetros uma lista de números. A função deve retornar a média dos elementos da lista. Uma professora ministra aula para duas turmas e deseja analisar compará-las. Construa um programa em que o usuário digita a quantidade de alunos em cada uma das duas turmas e as notas dos alunos de cada turma. Em seguida, seu programa deve mostrar, para cada turma, a média da turma, quantos alunos obtiveram nota acima desta média

da turma, quantos tem nota igual a média da turma e quantos tem nota abaixo da média da turma.

9. Mostre uma definição para a função `menor_distancia(x, y, lista_pontos)` que recebe as coordenadas (x, y) de um ponto no espaço cartesiano e uma lista de pontos, respectivamente. Uma lista `lista_pontos` com os pontos (x_1, y_1) , (x_2, y_2) e (x_3, y_3) deve ser da forma `[x1, y1, x2, y2, x3, y3]`. É importante observar que `lista_pontos` pode ter qualquer quantidade de pontos. A função deve retornar a distância de (x, y) para o ponto (x_i, y_i) em `lista_pontos` que está mais próximo de (x, y) . Em seguida, defina a função `vizinhos_mais_proximos(x, y, lista_pontos)` que deve retornar uma lista com todos os pontos de `lista_pontos` mais próximos de (x, y) . Caso apenas um ponto de `lista_pontos` seja mais próximo de (x, y) , sua função deve retornar uma lista apenas com as coordenadas desse ponto. Além disso, sua função deve retornar uma lista com o mesmo formato da lista `lista_pontos`.
10. Defina uma função `ordena_decrescente(lista)` que tem como parâmetro uma lista de números. A função deve ordenar, de forma decrescente, os elementos da lista e retornar a lista ordenada. Uma professora ministra aula para três turmas e deseja analisar as notas de cada turma em ordem decrescente. Apresente um programa em que o usuário digita a quantidade de alunos em cada uma das três turmas e as notas dos alunos de cada turma. Seu programa deve mostrar as notas de cada turma da maior para a menor.
11. Faça um programa que leia 10 valores e coloque em uma lista. Depois, mostre quais elementos da lista estão repetidos e quantas vezes cada um se repete.
12. Construa um programa que recebe 10 alturas e depois mostra a média das alturas e a altura que mais se repete.
13. Defina uma função recursiva `soma_quase_todos(lista, x, i)` que tem uma lista de números, um número e uma posição da lista como parâmetros, respectivamente. A função deve somar todos os elementos da lista por `x`, com exceção do elemento de índice `i`, e retornar a nova lista.
14. Usando recursão, defina a função `produto_listas(lista1, lista2)` que tem como parâmetros duas listas de números de mesmo tamanho. A função deve multiplicar os elementos correspondentes das duas listas e retornar a nova lista. Apresente um programa em que o usuário digita os preços de cada tipo de produto de uma loja. A quantidade de tipos de produtos deve ser definida pelo usuário. Depois, o usuário deve digitar a quantidade de produtos vendidos de cada tipo. Ao final da execução, seu programa deve mostrar a quantidade total de produtos vendidos e o valor total arrecadado com as vendas.
15. Defina a função `inverte_lista(lista)` recursivamente. A função tem uma lista como parâmetro e deve retornar uma nova lista com os elementos da original mas na ordem invertida. Por exemplo, `inverteLista([1,2,3])` deve retornar a lista `[3,2,1]`.

16. Defina recursivamente a função `palindromo(lista)` que tem como parâmetro uma lista. A função retorna `True` se lista é um palíndromo e retorna `False`, caso contrário. Uma lista é um palíndromo se é lida da mesma maneira da esquerda para a direita e da direita para a esquerda. Por exemplo, `[1,2,2,3,2,2,1]` é um palíndromo.