Fundamentos de Programação

Listas

Thiago Alves

Introdução

- Fazer um programa que recebe as duas notas de cada aluno de uma turma. O programa deve mostrar a média geral da turma e a quantidade de alunos com média maior ou igual a média da turma.

Listas em Python

```
l1 = []
l2 = [4, 5, 2]
l2[0] = input()
print l2[1]
l2.append(4)
l2[2] = 10
print range(4)
del l2[0]
```

Listas em Python

```
l1 = [5,2,4]
l2 = l1
l2[0] = 7
print l2 //[7,2,4]
print l1 //[7,2,4]
```

Exercício

- Faça um programa que recebe do usuário os elementos de duas listas com 10 elementos cada. O seu programa deve somar as duas listas e mostrar a lista resultante. A soma de listas é a soma das posições correspondentes

Exercício

- Faça um programa em que o usuário insere a quantidade de elementos de uma lista e o programa coloca o valor 0 nas posições pares e 1 nas posições ímpares.

Listas e Funções

```
def somaListas(lista1, lista2, tam):
  lista3 = [] #local
  for i in range(tam):
    lista3.append(lista1[i] + lista2[i])
  return lista3


listA = [1,2,3]
listB = [4,5,6]
lista = somaListas(listA, listB, 3)
print listA
print lista
```

Listas e Funções

```
def somaListas(lista1, lista2, tam):
  for i in range(tam):
    lista1[i] = lista1[i] + lista2[i]


listA = [1,2,3]
listB = [4,5,6]
somaListas(listA, listB, 3)
print listA #foi modificada
```

Listas e Funções

```
def somaListas(lista1, lista2, tam):
  lista3 = []
  for i in range(tam):
    lista3.append(lista1[i] + lista2[i])
  lista1 = [0,0]
  return lista3


listA = [1,2,3]
listB = [4,5,6]
print somaListas(listA, listB, 3)
print listA #nao foi modificada
```

Exercício

- Defina uma função `menorLista(lista, tam)` que possui como parâmetros uma lista de números e o tamanho da lista. A função deve retornar o menor elemento da lista. Faça um programa que recebe os salários dos funcionários de uma empresa e as horas trabalhadas pelos gerentes da empresa. Seu programa deve mostrar o menor salário dos funcionários e a menor quantidade de horas trabalhadas pelos gerentes.

Exercício

- Defina uma função `busca(lista, tam, elem)` que tem como parâmetros uma lista de números, o tamanho da lista e um número. A função deve retornar True se o elemento está na lista e False se o elemento não está na lista.

Exercício

- Defina uma função `somaLista(lista, tam)` que tem como parâmetros uma lista e o tamanho da lista. A função deve retornar a soma dos elementos da lista.

Exercício

- Defina uma função `qtdMaiorIgual(lista,tam, valor)` que possui uma lista, o tamanho da lista e um valor como parâmetros. A função deve retornar a quantidade de elementos da lista que é maior ou igual ao valor.

Exercício

- Fazer um programa que recebe as duas notas de cada aluno de uma turma. O programa deve mostrar a média geral da turma e a quantidade de alunos com média maior ou igual a média da turma. Use as funções `somaLista` e `qtdMaiorIgual` definidas anteriormente.

Exercício

- Uma loja tem 5 tipos de produtos diferentes e quer que você faça um programa para gerenciar os rendimentos. Cada produto deve receber um preço do usuário. Em cada venda, um comprador pode comprar qualquer quantidade de produtos de qualquer tipo. O programa deve mostrar o total de produtos vendidos, o total de vendas de cada tipo e o valor recebido no caixa no final do dia.

Exercício

- Faça um programa que recebe do usuário os elementos de um lista com 10 elementos e ordena a lista em ordem crescente.

Exercício

- Faça uma função `ordena(lista, tam)` que tem como parâmetros uma lista e o tamanho da lista. A função ordenar os elementos da lista.

Busca Binária

- Busca em uma lista ordenada sem precisar percorrer toda a lista:

```
def buscaBinaria(lista, tam, elem):
  inicio = 0
  final = tam-1
  while inicio <= final:
    posicao = inicio + (final - inicio)/2
    if lista[posicao] == elem:
      return True
    elif lista[posicao] < elem:
      inicio = posicao + 1
    else:
      final = posicao - 1
  return False
```