

# Mundo 3 - Listas

**1) Faça um programa que leia 5 valores numéricos e guarde-os em uma lista. No final, mostre qual foi o maior e o menor valor digitado e as suas respectivas posições na lista.**

Guanabara )

Neste código ele separou em 2 partes basicamente, uma para criar a lista e juntamente verificar qual o maior e qual o menor e a outra para verificar em quais posições estão os valores na lista, inclusive se o valor for igual.

- Declaramos a lista vazia
- Declaramos a variável e maior e menor valendo 0 ( sem fazer gambiarra )
- Estrutura de repetição para inserir os valores digitados pelo métodos append
- Após isso ele inicia o tratamento dos maiores e menores valores como inicialmente estão zerados, ele faz o primeiro if
  - Se o contador for zero ele vai inserir o mesmo valor para o maior e para o menor levando em consideração o valor da primeira posição da lista
  - Se não for a primeira posição ele vai para o else e verificar se a posição da lista for maior que o maior então ele recebe o valor e se a posição da lista for menor que o menor, ele recebe o valor da posição
- Prints apenas para firula
- Para saber em qual posição estão os números maiores fazemos um for para varrer novamente a lista e se o valor da posição da lista for igual ao maior, ele printa, da mesma forma para o menor valor

```
listanum = []
maior = menor = 0
for c in range(0,5):
    listanum.append(int(input(f'Digite o valor da posição {c}:')))
    if c == 0:
        maior = menor = listanum[c]
    else:
        if listanum[c] > maior:
            maior = listanum[c]
        if listanum[c] < menor:
            menor = listanum[c]
print('=-'*30)
print(f'Você digitou os valores {listanum}')
print(f'O maior número é o {maior} nas posições: ',end='')
for i, v in enumerate(listanum):
    if listanum[i] == maior:
        print(f'{i}...', end='')
print(f'\nO menor número é o {menor} nas posições: ',end='')
for i, v in enumerate(listanum):
    if listanum[i] == menor:
        print(f'{i}...', end='')
print(f'\n')
```

```

print()
print(f'O menor número é o {menor} nas posições: ',end='')
for i,v in enumerate(listanum):
    if listanum[i] == menor:
        print(f'{i}...',end='')

```

Se quiséssemos que fosse valores únicos, podemos criar primeiramente uma variável para armazenar o valor e depois dis

```

listanum = []
maior = menor = 0
for c in range(0,5):
    n = int(input(f'Digite um valor para a posição {c}: '))
    while True:
        if n not in listanum:
            listanum.append(n)
            break
        else:
            n = int(input(f'Valor já localizado, digite outro para a posição {c}: '))
    if c == 0:
        maior = menor = listanum[c]
    else:
        if listanum[c] > maior:
            maior = listanum[c]
        if listanum[c] < menor:
            menor = listanum[c]
print('=-'*30)
print(f'Você digitou os valores {listanum}')
print(f'O maior número é o {maior} nas posições: ',end='')
for i,c in enumerate(listanum):
    if listanum[i] == maior:
        print(f'{i}...', end='')
print()
print(f'O menor número é o {menor} nas posições: ',end='')
for i, c in enumerate(listanum):
    if listanum[i] == menor:
        print(f'{i}...',end='')

```

**2 ) Crie um programa onde o usuário possa digitar vários valores numéricos e cadastre-os em uma lista. Caso o número já exista lá dentro, ele não será adicionado. No final, serão exibidos todos os valores únicos digitados, em ordem crescente.**

Guanabara )

No exemplo do Guanabara, foi algo bastante simples:

- Declarando a lista vazia inicialmente, após isso puxa logo um loop infinito
- Declara n valendo o input inteiro
- Condicional se n não estiver em numeros ele faz o append e mostra que foi adicionado
- Se o numero estiver ele não adiciona e mostra a mensagem na tela
- Uma variável r é criada pelo input de s ou n
- Se ela estiver dentro de Nn ele faz o break
- Antes de finalizar ele organiza
- No final ele mostra o conteúdo da lista ordenada

```
numeros = list()
while True:
    n = int(input('Digite um valor: '))
    if n not in numeros:
        numeros.append(n)
        print('Numero adicionado com sucesso')
    else:
        print('Valor duplicado! Não vou adicionar...')
    r = str(input('Quer continuar [S/N] ?: '))
    if r in 'Nn':
        break
    print('-='*30)
numeros.sort()
print('Você digitou os números: {}'.format(numeros))
```

a)

A minha diferença é que fiz um teste lógico na opção de continuar ou não, fora isso o código permanece o mesmo.

```
numeros = []
while True:
    n = int(input('Digite um valor: '))
    if n not in numeros:
        numeros.append(n)
        print('Valor adicionado com sucesso...')
    else:
        print('Valor duplicado! Não vou adicionar...')
    r = input('Deseja continuar [S/N] ?: ')
    while r not in 'SsNn':
        r = input('Valor inválido! Deseja continuar [S/N] ?: ')
    if r in 'Nn':
        break
numeros.sort()
```

```
print('-='*30)
print(f'Você digitou os valores {numeros}')
```

**3 ) Crie um programa onde o usuário possa digitar cinco valores numéricos e cadastre-os em uma lista, já na opção correta de inserção ( sem usar o sort() ). No final, mostre a lista ordenada na tela.**

Guanabara )

Na solução do Guanabara, se formos ler o código passo a passo ficaria da seguinte forma:

- primeiro ele declarou uma lista vazia
- entrou com uma estrutura de repetição de 5 loops
- variável n recebe um input de valor inteiro
  - se contador for 0 ou se n for maior que o último valor da lista, então
  - numeros recebe o valor de n
  - print determinado

Senão:

- declarado uma variável chamada pos que servirá de contador e também para determinar a posição que vamos varrer a lista
- Enquanto pos for menor que o tamanho da lista
  - Se n for menor ou igual ao valor da posição pos na lista
  - Insert na posição pos o valor de n
  - print determinado
  - Break e sai do loop
  - pos recebe + 1 a cada loop

Então o código é dividido na primeira parte se o número for maior que a última posição

Se ele for menor, ele varrerá posição por posição até achar um que seja menor aí ele insere, lembrando que o insert coloca o valor na posição da lista e joga as que estavam para a direita.

```
numeros = list()
for c in range(0,5):
    n = int(input('Digite um valor: ')) # adicionando apenas se for maior q
    ue o ultimo ou o primeiro
    if c == 0 or n > numeros[-1]:
        numeros.append(n)
        print('Adicionado ao final da lista...')
    else:
        pos = 0
        while pos < len(numeros):      # varrendo e comparando todas as posi
```

ções

```
        if n <= numeros[pos]:
            numeros.insert(pos,n)
            print(f'Adicionado à posição {pos}...')
            break
        pos+=1
    print('Lista Atual ',numeros)
```

a )

Minha solução em macro funciona da mesma forma, a única diferença é que eu fiz a varredura das posições com outro for, dando certo e diminuindo 1 linha de código.

```
numeros = list()
for c in range(0,5):
    n = int(input('Digite um valor: ')) # adicionando apenas se for maior q
ue o ultimo ou o primeiro
    if c == 0 or n > numeros[-1]:
        numeros.append(n)
        print('Adicionado ao final da lista...')
    else:
        for i, c in enumerate(numeros): # varre posição por posição e se fo
r menor, insere
            if n <= c:
                numeros.insert(i,n)
                print(f'Adicionado à posição {i}...')
                break
print('-='*30)
print(f'Lista ordenada ---> {numeros}')
```

#### 4 ) Crie um programa que vai ler vários números e colocar em uma lista. Depois disso, mostre:

- Quantos números foram digitados
- A lista de valores, ordenada de forma decrescente
- Se o valor 5 foi digitado e está ou não na lista

Guanabara

```
valores = []
while True:
    valores.append(int(input('Digite um valor: ')))
    resp = str(input('Quer continuar? [S/N]: '))
```

```

    if resp in 'Nn':
        break
print('-='*30)
print('Você digitou {} elementos'.format(len(valores)))
valores.sort(reverse = True)
print('Os valores de forma decrescente são: {}'.format(valores))
if 5 in valores:
    print('O número 5 está presente na string')
else:
    print('O número 5 não se encontra na string')

```

a)

```

lista = []
while True:
    lista.append(int(input('Digite um número: ')))
    r = input('Quer continuar [S/N] ?: ')
    while r not in 'SsNn' or r=='':
        r = input('Valor inválido ! Quer continuar [S/N] ?: ')
    if r in 'Nn':
        break
print('-='*30)
print(f'Lista digitada --> {lista}')
print(f'No total foram digitados {len(lista)} valores')
lista.sort(reverse=True)
print(f'A lista de forma decrescente é --> {lista}')
if lista.count(5) > 0:
    print(f'O número 5 foi localizado {lista.count(5)} vezes')
else:
    print(f'O número 5 não foi localizado')

```

**5 ) Crie um programa que vai ler vários números e colocar em uma lista. Depois disso, crie duas listas extras que vão conter apenas os valores pares e os valores ímpares digitados, respectivamente. Ao final mostre o conteúdo das três listas geradas.**

Guanabara )

```

num = list()
pares = list()
impares = list()
while True:
    num.append(int(input('Digite um número: ')))

```

```

    resp = input('Quer continuar? [S/N]: ')
    if resp in 'Nn':
        break
for v in num:
    if v % 2 == 0:
        pares.append(v)
    else:
        impares.append(v)
print('='*30)
print(f'A lista completa e {num}')
print(f'A lista de pares e {pares}')
print(f'A lista de impares e {impares}')

```

a )

```

lista = list()
pares = list()
impares = list()
while True:
    n = int(input('Digite um valor: '))
    lista.append(n)
    if n % 2 == 0:
        pares.append(n)
    else:
        impares.append(n)
    resp = input('Deseja continuar [S/N] ?: ')
    while resp not in 'SsNn' or resp == '':
        resp = input('Valor inválido! Deseja continuar [S/N]?: ')
    if resp in 'Nn':
        break
print('='*30)
print(f'Lista digitada --> {lista}')
print(f'Lista de números pares --> {pares}')
print(f'Lista de números ímpares --> {impares}')

```

**6 ) Crie um programa onde o usuário digite uma expressão qualquer que use parênteses. Seu aplicativo deverá analisar se a expressão passada está com os parênteses abertos e fechados na ordem correta.**

Guanabara )

```

expr = input('Digite a expressão: ')
pilha = []
for simb in expr:
    if simb == '(':
        pilha.append('(')

```

```

        elif simb == ')':
            if len(pilha) > 0:
                pilha.pop()
            else:
                pilha.append(')')
                break
    if len(pilha) == 0:
        print('Sua expressão está válida')
    else:
        print('Sua expressão está inválida')

```

a ) A solução do Guanabara foi por lista, eu fiz por contagem utilizando uma variável de saldo.

```

lista = []
exp = input('Dgite sua expressão: ')
paren = 0
for c in exp:
    if c == '(':
        paren += 1
    elif c == ')':
        paren -= 1
if paren == 0:
    print('Expressão válida.....')
else:
    print('Expressão inválida!')
    if paren > 0:
        print(f'Ficou faltando fechar {paren} parentesis')
    else:
        print(f'Estão com {paren*-1} parentesis a mais')

```

b)

```

lista = []
exp = input('Dgite sua expressão: ')
paren = []
for c in exp:
    if c == '(':
        paren.append(c)
    elif c == ')':
        paren.pop()
if len(paren) == 0:
    print('Expressão válida.....')
else:
    print('Expressão inválida!')
    if len(paren) > 0:

```



```
    print(f'Ficou faltando fechar {len(paren)} parentesis')
else:
    print(f'Estão com {len(paren)*-1} parentesis a mais')
```