**Logotipo

Descrição gerada automaticamente com confiança baixa**

**Disciplina:** Paradigmas de Linguagens de Programação em Python

**Docente:** Heleno Filho

**Aluno:** José Gabriel da Silva Batista

**Lista de Exercícios de Algoritmo Repetição e Vetores – UFBA**

**1.**

numeros = []

for i in range(10):

numero = int(input(f"Digite o {i+1}º número inteiro: "))

numeros.append(numero)

print("Lista de números:", end=" ")

for numero in numeros:

print(numero, end=" ")

**2.**

numeros = []

for i in range(10):

numero = int(input(f"Digite o {i+1}º número inteiro: "))

numeros.append(numero)

numero\_verificar = int(input("Digite um número para verificar na lista: "))

if numero\_verificar in numeros:

print(f"O número {numero\_verificar} está na lista.")

else:

print(f"O número {numero\_verificar} não está na lista.")

**3.**

numeros = []

positivos = []

negativos = []

for i in range(10):

numero = float(input(f"Digite o {i+1}º número real: "))

numeros.append(numero)

if numero > 0:

positivos.append(numero)

elif numero < 0:

negativos.append(numero)

media = sum(numeros) / len(numeros)

maior = max(numeros)

menor = min(numeros)

print(f"Média dos elementos: {media:.2f}")

print(f"Maior elemento: {maior:.2f}")

print(f"Menor elemento: {menor:.2f}")

print(f"Quantidade de elementos positivos: {len(positivos)}")

print(f"Quantidade de elementos negativos: {len(negativos)}")

**4.**

reajustadas

nomes = []

idades = []

indenizacoes\_reajustadas = []

while True:

nome = input("Digite o nome completo do paciente (ou 'sair' para encerrar): ")

if nome.lower() == 'sair':

break

idade = int(input("Digite a idade do paciente: "))

valor\_base = float(input("Digite o valor base de indenização: "))

if idade <= 12:

reajuste = 0.30

elif 13 <= idade <= 49:

reajuste = 0.10

elif 50 <= idade <= 65:

reajuste = 0.15

else:

reajuste = 0.35

indenizacao\_reajustada = valor\_base \* (1 + reajuste)

nomes.append(nome)

idades.append(idade)

indenizacoes\_reajustadas.append(indenizacao\_reajustada)

print("Resultados:")

for i in range(len(nomes)):

print(f"Nome: {nomes[i]}, Idade: {idades[i]}, Indenização Reajustada: R${indenizacoes\_reajustadas[i]:.2f}")

**5.**

import math

def calcular\_norma(vetor):

norma = math.sqrt(sum(x \*\* 2 for x in vetor))

return norma

def calcular\_soma(vetor1, vetor2, vetor3):

soma = [x + y + z for x, y, z in zip(vetor1, vetor2, vetor3)]

return soma

N = int(input("Digite o tamanho dos vetores: "))

vetor1 = []

vetor2 = []

vetor3 = []

print("Digite os valores do vetor 1:")

for i in range(N):

valor = float(input(f"Digite o valor {i+1}: "))

vetor1.append(valor)

print("Digite os valores do vetor 2:")

for i in range(N):

valor = float(input(f"Digite o valor {i+1}: "))

vetor2.append(valor)

print("Digite os valores do vetor 3:")

for i in range(N):

valor = float(input(f"Digite o valor {i+1}: "))

vetor3.append(valor)

norma1 = calcular\_norma(vetor1)

norma2 = calcular\_norma(vetor2)

norma3 = calcular\_norma(vetor3)

maior\_norma = max(norma1, norma2, norma3)

soma\_vetores = calcular\_soma(vetor1, vetor2, vetor3)

print(f"Norma do vetor 1: {norma1:.2f}")

print(f"Norma do vetor 2: {norma2:.2f}")

print(f"Norma do vetor 3: {norma3:.2f}")

if norma1 == maior\_norma:

print("O vetor 1 tem a maior norma.")

elif norma2 == maior\_norma:

print("O vetor 2 tem a maior norma.")

else:

print("O vetor 3 tem a maior norma.")

print(f"Vetor Soma dos três vetores: {soma\_vetores}")

**6.**

clientes = []

while True:

nome = input("Digite o nome completo do cliente (ou 'sair' para encerrar): ")

if nome.lower() == 'sair':

break

rg = input("Digite o RG do cliente: ")

cpf = input("Digite o CPF do cliente: ")

telefone = input("Digite o telefone do cliente: ")

cliente = [nome, rg, cpf, telefone]

clientes.append(cliente)

print("Cadastro de Clientes:")

for cliente in clientes:

print(f"Nome: {cliente[0]}\nRG: {cliente[1]}\nCPF: {cliente[2]}\nTelefone: {cliente[3]}\n")

**7.**

elementos = []

for i in range(10):

elemento = int(input(f"Digite o {i+1}º elemento inteiro: "))

elementos.append(elemento)

valores\_pares = sum(1 for elemento in elementos if elemento % 2 == 0)

print(f"Quantidade de valores pares na lista: {valores\_pares}")

**8.**

numeros\_pares = []

numeros\_impares = []

tamanho\_maximo = 5

while True:

valor = int(input("Digite um valor inteiro (ou 0 para encerrar a leitura): "))

if valor == 0:

break

if valor % 2 == 0:

numeros\_pares.append(valor)

else:

numeros\_impares.append(valor)

if len(numeros\_pares) == tamanho\_maximo:

print("Lista de números pares:", numeros\_pares)

numeros\_pares = []

if len(numeros\_impares) == tamanho\_maximo:

print("Lista de números ímpares:", numeros\_impares)

numeros\_impares = []

if numeros\_pares:

print("Lista de números pares restante:", numeros\_pares)

if numeros\_impares:

print("Lista de números ímpares restante:", numeros\_impares)

**9.**

gabarito = input("Digite o gabarito da prova com 20 questões (exemplo: ABCDEABCDEABCDEABCDE): ")

gabarito = list(gabarito)

resultados\_alunos = []

num\_alunos = 50

acertos\_necessarios = 6 \* 2 # Cada questão correta vale 0,5 ponto

for aluno in range(1, num\_alunos + 1):

nome = input(f"Digite o nome do aluno {aluno}: ")

respostas = input(f"Digite as respostas do aluno {aluno} (exemplo: ABCDEABCDEABCDEABCDE): ")

respostas = list(respostas)

num\_acertos = sum(1 for g, r in zip(gabarito, respostas) if g == r)

nota\_final = num\_acertos \* 0.5

if nota\_final >= acertos\_necessarios:

resultado = "APROVADO"

else:

resultado = "REPROVADO"

resultados\_alunos.append((nome, num\_acertos, resultado))

for nome, num\_acertos, resultado in resultados\_alunos:

print(f"Aluno: {nome}, Número de Acertos: {num\_acertos}, Resultado: {resultado}")

**10.**

gabarito = input("Digite o gabarito da prova com 20 questões (exemplo: ABCDEABCDEABCDEABCDE): ")

gabarito = list(gabarito)

resultados\_alunos = []

num\_alunos = 50

acertos\_necessarios = 6 \* 2 # Cada questão correta vale 0,5 ponto

for aluno in range(1, num\_alunos + 1):

nome = input(f"Digite o nome do aluno {aluno}: ")

respostas = input(f"Digite as respostas do aluno {aluno} (exemplo: ABCDEABCDEABCDEABCDE): ")

respostas = list(respostas)

num\_acertos = sum(1 for g, r in zip(gabarito, respostas) if g == r)

nota\_final = num\_acertos \* 0.5

if nota\_final >= acertos\_necessários:

resultado = "APROVADO"

else:

resultado = "REPROVADO"

resultados\_alunos.append((nome, nota\_final, resultado))

for nome, nota\_final, resultado in resultados\_alunos:

print(f"Aluno: {nome}, Nota Final: {nota\_final}, Resultado: {resultado}")

**11.**

numeros\_reais = []

for i in range(15):

valor = float(input(f"Digite o {i+1}º número real: "))

numeros\_reais.append(valor)

while True:

codigo = int(input("Digite um código (1 para ordem direta, 2 para ordem inversa, 0 para encerrar): "))

if codigo == 0:

break

elif codigo == 1:

print("Lista na ordem direta:", numeros\_reais)

elif codigo == 2:

print("Lista na ordem inversa:", numeros\_reais[::-1])

else:

print("Código inválido. Use 0, 1 ou 2 para escolher a ação.")

**12.**

a = int(input("Digite o valor de 'a': "))

b = int(input("Digite o valor de 'b': "))

N = int(input("Digite o tamanho da lista: "))

lista = []

for i in range(N):

elemento = int(input(f"Digite o {i+1}º elemento da lista: "))

lista.append(elemento)

contagem = sum(1 for elemento in lista if a <= elemento <= b)

print(f"A quantidade de elementos no intervalo [{a}, {b}] é: {contagem}")