

2º Trabalho de Int. a Programação

Discente: Maria Letícia Gomes de Matos

1-

```
V = int(input("Insira um valor inteiro da base de potência: "))
N = int(input("Insira um valor inteiro para o expoente: "))
P = 0

for i in range (N, 0, -1):
    termo_atual = (V ** i) - i
    P = P + termo_atual

print("UFC - PROGRAMAÇÃO")
print("Nome: Leticia")
print("Valor de V: ", V)
print("Valor de N: ", N)
print("Valor de P: ", P)
```

2-

```
frase = str(input("Digite uma frase: ").upper().replace(" ", ""))
frase_inversa = "".join(reversed(frase))
print ("Você digitou a frase: ", frase)
print (frase_inversa)

if frase == frase_inversa:
    print("A frase é um palíndromo!")
else:
    print("A frase não é um palíndromo.")
```

3-

```
DD = 5
MM = 2
AA = 2006

if MM >= 9 and DD >= 15:
    DD = 15
    ID = 2024 - AA
    MM = 12 - (MM - 9 -1)
elif MM >= 9 and DD < 15:
    DD = 15 - DD
    ID = 2024 - AA
    MM = 12 - (MM - 9 - 1)
elif MM < 9 and DD >= 15:
    ID = 2024 - AA - 1
    MM = 12 - (9 - MM)
    DD = 15
elif MM < 9 and DD < 15:
    ID = 2024 - AA - 1
    MM = 12 - (9 - MM - 1)
    DD = 15 - DD

ID = ID * 365 + MM * 30 + DD
print(ID)
```

4 -

- a. Serve para tomar decisões, especificando qual o caminho a ser seguido pelo computador para realizar a execução do código.
- b. A estrutura básica é o **if**.

Exemplo 1:

```
idade = 18
if idade >= 18:
    print("Você é maior de idade")
```

Exemplo 2:

```
N = 5
if N > 0:
    print("N é um número positivo")
```

- c. Uma variação da estrutura seria o **else**

Exemplo 1:

```
N = 10
if N > 0:
    print("N é um número positivo")
else:
```

```
print("N é um número negativo")
```

Exemplo 2:

```
numero = 12
if numero % 2 == 0:
    print("O número é par")
else:
    print("O número é ímpar")
```

d. Exemplo 1:

```
tamanho = "G"

if tamanho == "P":
    print("Pequeno")
elif tamanho == "M":
    print("Médio")
else:
    print("Grande")
```

Exemplo 2:

```
cor = "amarelo"
```

```
if cor == "verde":
    print("Siga")
elif cor == "amarelo":
    print("Atenção")
else:
    print("Pare")
```

e. Um exemplo pode ser o de restrição de altura e idade para entrar em um brinquedo, como por exemplo:

```
idade = 5
altura = 1.45
```

```
if idade >= 5
print("Idade ok")
if altura >= 1.40
print("Pode entrar no brinquedo!")
else:
```

```
print("Barrado pela altura")
else:
    print("Barrado pela idade")
```

5 -

```
print("UFC _ Programação")
print("Nome: Letícia\n")
print("Sequência de Fibonacci")
print("-" * 30)
n = int(input("Insira um número N para definir quantidade de números a ser apresentada: \n"))
print("-" * 30)
t1 = 0
t2 = 1
cont = 0
fib = []
fib.append(t1)
fib.append(t2)

while cont < n - 2:
    t3 = t1 + t2
    t1 = t2
    t2 = t3
    fib.append(t3)
    cont += 1

for i in range(1, len(fib) + 1):
    for j in range(i):
        print(fib[j], end=" ")
    print()

for i in range(len(fib) - 1, 0, -1):
    for j in range(i):
        print(fib[j], end=" ")
    print()
```

6 -

```
A = int(input("Insira o valor de A: "))
B = int(input("Insira o valor de B: "))

F = 0

for x in range(1, B + 1):
    F += ((A - (x - 1)) ** (B - (x - 1))) / x

print("UFC - PROGRAMAÇÃO")
print("Curso: Química")
print("Nome: Letícia")
print("Valor de A: ", A)
print("Valor de B: ", B)
print("Valor da Função F: ", F)
```

7 -

```
VrLim = int(input("Digite o valor limite: "))
RZ = 3
VrSeq = 0
print("UFC - Programação")
print("Nome: Letícia Matos")
print("Programa Questão 7\n")

while VrLim > VrSeq:
    print(VrSeq)
    VrSeq += RZ
    if RZ == -1:
        RZ = 3
    else:
        RZ = -1

print("Valor Limite = ", VrLim)
```

8 -

```

print("UFC - PROGRAMAÇÃO")
print("Curso: Química")

n = int(input("Insira um valor inteiro para N: "))
subtrator = 1
sequencia = []
while n >=0:
    sequencia.append(n)
    n = (n - subtrator)
    subtrator += 1

print("Valor de N: ", sequencia[0])
💡
print("Valor de sequencia: |", sequencia)

```

9 -

```

print("Curso: Química")
print("Nome: Letícia")

M = int(input("Digite o número de linhas (M): "))
N = int(input("Digite o número de colunas (N): "))

primeira = []
print("\nDigite os elementos da Primeira Matriz:")
for i in range(M):
    linha = input(f"Linha {i+1}: ").split()
    primeira.append(linha)

segunda = []
for i in range(M - 1, -1, -1):
    linha_invertida = []
    for j in range(N - 1, -1, -1):
        linha_invertida.append(primeira[i][j])
    segunda.append(linha_invertida)

print("\nUFC - Programação")
print(f"Curso: {curso}")
print(f"Nome: {nome}\n")

print("Segunda Matriz: M x N")
for linha in segunda:
    print(" ".join(linha))

```

```

print("\nPrimeira Matriz: M x N")
for linha in primeira:
    print(" ".join(linha))

```

10 -

```

N = int(input("Digite o número de elementos: "))
vetor1 = []
vetor2 = []

for i in range(N):
    vetor1.append(int(input(f"Digite o elemento {i+1} do Vetor1: ")))

for i in range(N):
    vetor2.append(int(input(f"Digite o elemento {i+1} do Vetor2: ")))

resultados = []

for i in range(N):
    diferença = vetor1[i] - vetor2[N - 1 - i]
    resultados.append(diferença)

print("\nUFC - Programação")
print("Resultado de Todas as Subtrações:")
print(" ".join(str(x) for x in resultados))

print(f"\nNúmero de Elementos = {N}")
print(f"Elementos do Vetor1: {vetor1}")
print(f"Elementos do Vetor2: {vetor2}")

print("\nFim do Trabalho")

```

11 -

```

precos = []
for i in range(8):
    preco = float(input(f"Digite o preço do produto {i+1}: "))
    precos.append(preco)

estoque = []
for i in range(6):
    linha = []
    print(f"\nArmazém {i+1}:")
    for j in range(8):
        qtd = int(input(f"Quantidade do produto {j+1}: "))
        linha.append(qtd)
    estoque.append(linha)

print("\nQuantidade total em cada armazém:")
for i in range(6):
    total = 0
    for j in range(8):
        total += estoque[i][j]
    print(f"Armazém {i+1}: {total}")

print("\nQuantidade total de cada produto:")
for j in range(8):
    maior = estoque[0][j]
    prod = 0
    arm = 0
    for i in range(6):
        if estoque[i][j] > maior:
            maior = estoque[i][j]
            prod = i + 1
            arm = j + 1
    print(f"Produto {j+1}: {maior} ({prod}, {arm})")

```

12 -

```

conj = []
quant = 0
somatorio = 0
print("UFC - Programação")
print("Nome: Letícia")
print("----Iniciando digitação (digite 0 para sair)----")

while True:
    num = int(input("Digite números inteiros: "))
    if num == 0:
        break
    quant += 1
    somatorio += num
    conj.append(num)

print("~*30)
print("Conjunto: ", conj)
print("Somatório dos valores: ", somatorio)
print("Quantidade de números informados: ", quant)
print("~*30)

```

13 -

```

conj = []
quant = 0
maiorpar = 0

n = int(input("Insira um número inteiro e positivo para representar a quantidade de elementos: "))

for i in range(n):
    elem = int(input("Insira um número inteiro para o conjunto: "))
    conj.append(elem)
    quant += 1
    if elem > maiorpar and elem % 2 == 0:
        maiorpar = elem

print("UFC - Programação")
print("Curso: Química")
print("Nome: Letícia")
print("-*30)
print("Números de elementos: ", quant)
print("Maior valor par: ", maiorpar)
print("Posição do maior valor par: ", conj.index(maiorpar) + 1)
print("Elementos do conjunto: ", conj)

```

14 -

```

primos = []
print("UFC - PROGRAMAÇÃO")
n = int(input("Insira o valor limite: "))

for num in range(1, n + 1):
    divisores = 0
    for i in range(1, num + 1):
        if num % i == 0:
            divisores += 1
    if divisores == 2:
        primos.append(num)

print("-" * 30)
print("Números primos:", primos)

```

15 -

- a. Executar um bloco de código várias vezes, enquanto uma condição for verdadeira ou por um número definido de vezes.

b. `for i in range(5):
 print(i)`

```

i = 0
while i < 5:
    print(i)
    i += 1

```

```

for x in [1, 2, 3]:
    print(x)

```

- c. `for i in range(3):
 for j in range(2):
 print(i, j)`

16 -

```
inicio = int(input("Valor de Início: "))
fim = int(input("Valor de Fim: "))
N = int(input("Número de Elementos: "))

elementos = []
somatorio = 0

for i in range(N):
    valor = int(input(f"Elemento {i+1}: "))
    elementos.append(valor)

    if valor % 2 == 0 and inicio <= valor <= fim:
        somatorio += valor

print("\nUFC - Programação")
print(f"Número de Elementos = {N}")
print(f"Somatório dos Pares = {somatorio}")
print(f"Valor de Início = {inicio}")
print(f"Valor de Fim = {fim}")
print(f"Elementos: {elementos}")
```

17 -

```
import random
nome = input("Nome: ")
matricula = input("Matrícula: ")
vetor = random.sample(range(1, 2001), k=1000)
vetor.sort()
valor = int(input("Valor pesquisado: "))

inicio = 0
fim = len(vetor) - 1
comparacoes = 0
posicao = -1

while inicio <= fim:
    comparacoes += 1
    meio = (inicio + fim) // 2
    if vetor[meio] == valor:
        posicao = meio
        break
    elif valor < vetor[meio]:
        fim = meio - 1
    else:
        inicio = meio + 1

print("\nUFC - Programação")
print(f"Nome: {nome}")
print(f"Matrícula: {matricula}\n")
```

```

print("Elementos com a Posição:")
for i, v in enumerate(vetor):
    print(f"{i}:{v}", end=" ")
    if (i + 1) % 10 == 0:
        print()

print(f"\n\nValor Pesquisado: {valor}")

if posicao != -1:
    print(f"Posição: {posicao}")
else:
    print("Posição: Valor não encontrado")

print(f"Número de Comparações: {comparacoes}")

```

18 -

```

N = int(input("Digite o número de linhas: "))

triangulo = []

for i in range(N):
    if i == 0:
        triangulo.append([1])
    else:
        linha_anterior = triangulo[i - 1]
        nova_linha = [1]

        for j in range(len(linha_anterior) - 1):
            nova_linha.append(linha_anterior[j] + linha_anterior[j + 1])

        nova_linha.append(1)
        triangulo.append(nova_linha)

for linha in triangulo:
    print(*linha)

for linha in triangulo[-2::-1]:
    print(*linha)

```

19 -

```
print("UFC - Programação")
print("Nome: Letícia")

n = int(input("Número de Elementos = "))
conjunto = []

print("Digite os elementos:")
for k in range(n):
    num = int(input(f"Elemento {k}: "))
    conjunto.append(num)

soma_orig = 0
qtd_orig = 0

for i in range(n):
    valor = conjunto[i]

    if (i % 2 == 0) and (valor % 2 != 0):
        soma_orig += valor
        qtd_orig += 1

conjunto_inv = conjunto[::-1]

soma_inv = 0
qtd_inv = 0
```

```
| for i in range(n):
|     valor = conjunto_inv[i]
|
|     if (i % 2 == 0) and (valor % 2 != 0):
|         soma_inv += valor
|         qtd_inv += 1
|
| print()
| print(f"Elementos do Conjunto: {conjunto}")
| print(f"Somatório (índices pares e valores ímpares): {soma_orig}")
| print(f"Quantidade = {qtd_orig}")
|
| print()
| print(f"Elementos Conjunto Inverso: {conjunto_inv}")
| print(f"Somatório (índices pares e valores ímpares): {soma_inv}")
| print(f"Quantidade = {qtd_inv}")
```

20 -

```
conj = []
quant = 0
maiorvalor = 0

n = int(input("Insira um número inteiro e positivo para representar a quantidade de elementos:"))

for i in range(n):
    elem = int(input("Insira um número inteiro para o conjunto:"))
    conj.append(elem)
    quant += 1
    if elem > maiorvalor:
        maiorvalor = elem

print("UFC - Programação")
print("Curso: Química")
print("Nome: Letícia")
print("-"*30)
print("Números de elementos: ", quant)
print("Maior valor: ", maiorvalor)
print("Posição do maior valor: ", conj.index(maiorvalor) + 1)
print("Elementos do conjunto: ", conj)
```

21 -

```
string1 = input("Digite o 1º nome: ")
string2 = input("Digite o 2º nome: ")

tam1 = len(string1)
tam2 = len(string2)

posicao = string1.find(string2)

caractere = input("Digite um caractere para contagem na 1º string: ")
quantidade = string1.count(caractere)

string1_maiuscula = string1.upper()
string2_minuscula = string2.lower()

concat_1_2 = string1 + string2
concat_2_1 = string2 + string1

print("\nUFC - Programação")
print("Utilizando String:\n")

print(f"Informação da 1º string = {string1}")
print(f"Informação da 2º string = {string2}\n")

print(f"Item (i) = {tam1} caracteres na 1º string e {tam2} caracteres na 2º string")
```

```
if posicao != -1:
    print(f"Item (ii) = A 2º string inicia na posição {posicao} da 1º string")
else:
    print("Item (ii) = A 2º string NÃO é substring da 1º string")

print(f"Item (iii) = O caractere '{caractere}' aparece {quantidade} vez(es) na 1º string")
print(f"Item (iv) = {string1_maiuscula}")
print(f"Item (v) = {string2_minuscula}")
print(f"Item (vi) = {concat_1_2}")
print(f"Item (vii) = {concat_2_1}")
```