

VITOR ARANTES 2º Período

## ALGORITMO 481

```
def converter_base(numero, base):
```

```
    if not 2 <= base <= 10:
```

```
        return "A base deve estar entre 2 e 10"
```

```
    if numero == 0:
```

```
        return "0"
```

```
    resultado = ""
```

```
    while numero > 0:
```

```
        digito = numero % base
```

```
        resultado = str(digito) + resultado
```

```
        numero //= base
```

```
    return resultado
```

```
# Exemplo de uso:
```

```
numero_base_10 = 42
```

```
nova_base = 5
```

```
resultado = converter_base(numero_base_10, nova_base)
```

```
print(f"O número {numero_base_10} na base {nova_base} é igual a {resultado}")
```

## ALGORITMO 483

```
def produto_interno(vetor1, vetor2, tamanho):  
    if len(vetor1) < tamanho or len(vetor2) < tamanho:  
        return "Os vetores não têm tamanho suficiente para a quantidade especificada de  
        elementos."  
  
    resultado = 0  
  
    for i in range(tamanho):  
        resultado += vetor1[i] * vetor2[i]  
  
    return resultado  
  
# Exemplo de uso:  
vetor_a = [1, 2, 3]  
vetor_b = [4, 5, 6]  
quantidade_elementos = 3  
resultado_produto_interno = produto_interno(vetor_a, vetor_b, quantidade_elementos)  
print(f"O produto interno dos vetores {vetor_a} e {vetor_b} é {resultado_produto_interno}")
```

## ALGORITMO 485

```
def imprimir_caracteres(vetor_inteiros, vetor_caracteres, tamanho):  
    if len(vetor_inteiros) < tamanho or len(vetor_caracteres) < tamanho:  
        print("Os vetores não têm o tamanho especificado.")  
        return  
  
    for i in range(tamanho):  
        n = vetor_inteiros[i]  
        char = vetor_caracteres[i]
```

```
for _ in range(n):  
    print(char, end="")
```

```
print() # Pule para a próxima linha após imprimir os caracteres n vezes
```

```
# Exemplo de uso:
```

```
vetor_inteiros = [2, 3, 1, 4]
```

```
vetor_caracteres = ['A', 'B', 'C', 'D']
```

```
tamanho = 4
```

```
imprimir_caracteres(vetor_inteiros, vetor_caracteres, tamanho)
```

## ALGORITMO 487

```
def inverte(vetor, tamanho):
```

```
    if tamanho < 2:
```

```
        return vetor # Nada a inverter
```

```
    meio = tamanho // 2
```

```
    for i in range(meio):
```

```
        vetor[i], vetor[tamanho - 1 - i] = vetor[tamanho - 1 - i], vetor[i]
```

```
# Exemplo de uso:
```

```
vetor = [1, 2, 3, 4, 5]
```

```
tamanho = len(vetor)
```

```
inverte(vetor, tamanho)
```

```
print(vetor)
```

## ALGORITMO 490

```
def substituir_caractere(vetor, tamanho, caractere):
```

```
total_retirados = 0
```

```
for i in range(tamanho):
```

```
    if vetor[i] == caractere:
```

```
        vetor[i] = '*'
```

```
        total_retirados += 1
```

```
return total_retirados
```

```
# Exemplo de uso:
```

```
vetor = ['a', 'b', 'c', 'a', 'd', 'e', 'a']
```

```
tamanho = len(vetor)
```

```
caractere_a_substituir = 'a'
```

```
total_retirados = substituir_caractere(vetor, tamanho, caractere_a_substituir)
```

```
print("Vetor após substituição:", vetor)
```

```
print("Total de caracteres retirados:", total_retirados)
```

## ALGORITMO 493

```
def verificar_ordem(vetor):
```

```
    crescente = True
```

```
    decrescente = True
```

```
    for i in range(1, len(vetor)):
```

```
        if vetor[i] > vetor[i - 1]:
```

```
            decrescente = False
```

```
        elif vetor[i] < vetor[i - 1]:
```

```
            crescente = False
```

```
    if crescente:
```

```
        return "O vetor está ordenado de forma crescente."
```

elif decrescente:

    return "O vetor está ordenado de forma decrescente."

else:

    return "O vetor não está ordenado."

# Exemplo de uso:

vetor\_crescente = [1, 2, 3, 4, 5]

vetor\_decrescente = [5, 4, 3, 2, 1]

vetor\_desordenado = [3, 1, 4, 2, 5]

print(verificar\_ordem(vetor\_crescente))

print(verificar\_ordem(vetor\_decrescente))

print(verificar\_ordem(vetor\_desordenado))

## ALGORITMO 494

def troca(vetor, indice1, indice2):

    vetor[indice1], vetor[indice2] = vetor[indice2], vetor[indice1]

def ordenar\_vetor(vetor):

    tamanho = len(vetor)

    for i in range(tamanho):

        for j in range(0, tamanho-i-1):

            if vetor[j] > vetor[j+1]:

                troca(vetor, j, j+1)

# Exemplo de uso:

vetor = [5, 2, 9, 1, 5, 6]

ordenar\_vetor(vetor)

print("Vetor ordenado:", vetor)

## ALGORITMO 495

```
def ordenar_vetor_caracteres(vetor):  
    return sorted(vetor)  
  
# Exemplo de uso:  
vetor = ['z', 'a', 'c', 'b', 'f', 'e']  
vetor_ordenado = ordenar_vetor_caracteres(vetor)  
print("Vetor ordenado:", vetor_ordenado)
```

## ALGORITMO 497

```
def busca_binaria(vetor, tamanho, chave):  
    esquerda, direita = 0, tamanho - 1  
  
    while esquerda <= direita:  
        meio = (esquerda + direita) // 2  
  
        if vetor[meio] == chave:  
            return meio # Encontramos a chave, retorna a posição  
        elif vetor[meio] < chave:  
            esquerda = meio + 1  
        else:  
            direita = meio - 1  
  
    return -1 # A chave não foi encontrada no vetor  
  
# Exemplo de uso:  
vetor_ordenado = [1, 2, 3, 4, 5, 6, 7, 8, 9]  
tamanho = len(vetor_ordenado)  
chave = 5  
  
posicao = busca_binaria(vetor_ordenado, tamanho, chave)
```

```
if posicao != -1:
    print(f"A chave {chave} foi encontrada na posição {posicao}.")
else:
    print(f"A chave {chave} não foi encontrada no vetor.")
```

## ALGORITMO 499

```
# Função para ler dados do vetor
```

```
def ler_dados_vetor(vetor):
    print("Digite os dados do vetor:")
    for i in range(5):
        vetor[i] = int(input(f"Digite o valor para a posição {i + 1}: "))
```

```
# Função para ordenar o vetor
```

```
def ordenar_vetor(vetor):
    vetor.sort()
```

```
# Função para imprimir o vetor
```

```
def imprimir_vetor(vetor):
    print("Vetor:", vetor)
```

```
# Vetor com dimensão 5
```

```
vetor = [0, 0, 0, 0, 0]
```

```
while True:
```

```
    print("\nMENU:")
    print("1 - Dados do VETOR")
    print("2 - Ordena VETOR")
    print("3 - Imprime VETOR")
    print("4 - Sai do programa")
```

```
    opcao = input("OPÇÃO: ")
```

```
if opcao == '1':  
    ler_dados_vetor(vetor)  
elif opcao == '2':  
    ordenar_vetor(vetor)  
    print("Vetor ordenado.")  
elif opcao == '3':  
    imprimir_vetor(vetor)  
elif opcao == '4':  
    print("Programa encerrado.")  
    break  
else:  
    print("Opção inválida. Por favor, escolha uma opção válida.")
```