



UNIVERSIDADE
VILA VELHA
ESPÍRITO SANTO

Programação de Computadores
Linguagem C de Programação

Conceitos básicos:

O que vamos estudar hoje:

- Conceitos: **Vetor (Array)**: é uma **Estrutura de Dados** usada para memorizar (memória primária) grandes volumes de dados.
- Características: **Vetor: 1D (Dimensão: Linha), Matriz (Tabela): 2D (Dimensão: Linha e Coluna) e Prisma: 3D (Dimensão: X, Y e Z - Computação Gráfica)**
- Características: **Homogêneo (mesmo tipo: inteiro ou real) e Estático (Tamanho Fixo)**
- Componentes de um Vetor: **Elementos ou termos, Indexação (Índice) e Tamanho Máximo (Fixo)**.
- **Indexação (Índice)**: É uma variável inteira usada para acessar os termos ou elementos de um vetor (Valores).
- **Indexação (Índice Crescente)**: Índices entre [0 ... TAMANHO - 1]
- **Indexação**: Posição = Índice + 1
- **Indexação**: Índice = Posição - 1
- **Tipos de dados**: INT ou FLOAT ou CHAR

Exemplo de Vetor: Notas do 1º Bimestre

POSIÇÃO NA...	Aluno 1	Aluno 2	Aluno 3	...	Aluno 48	Aluno 49	A
BIMESTRE 1	7.7	5.5	9.2	...	5.7	6.3	
ÍNDICE (cres...	0	1	2	...	47	48	

▼ ARRAY FLOAT

▼ Exemplo 1:

Ler a nota turma: 1º Bimestre de 50 alunos e armazenar os dados em um vetor.

Após isso, faça:

LETRA A) Qual a média da turma.

Link reply.it: [Aqui!](#)

LETRA B) Quantidade em valores absolutos e relativos (%) de alunos:

- Acima (e inclusive) da média da turma.
- Abaixo da média da turma.

Link reply.it: [Aqui!](#)

LETRA C) Exibir na tela o Relatório:

- Nota dos alunos (e sua respectiva posição) dos alunos acima (e inclusive) da média da turma.
- Nota dos alunos (e sua respectiva posição) dos alunos abaixo da média da turma.

Link reply.it: [Aqui!](#)

▼ ARRAY INT

Como gerar números aleatorios? [Link](#)

Vamos usar as funções:

- RAND: [Link](#) - [EXEMPLO](#)
- SRAND: [Link](#)

▼ Exemplo 2:

Preencher um vetor (SIZE = 1000) com senhas aleatórias (rand + srand) de 4 dígitos: [1000, 9999].

Após isso, faça:

LETRA A: Enumerar e exibir na tela a quantidade de senhas fortes, com a seguinte criptografia:

-> SENHA FORTE: Ímpar E Múltiplas de 39.

Link reply.it: [Aqui!](#)

LETRA B: Exibir na tela a Menor e a Maior senha sorteada com a criptografia acima.

Link reply.it: [Aqui!](#)

▼ Exemplo 3:

Escrever um algoritmo que preencha um vetor (SIZE = 10) com valores aleatórios entre [0, 9].

Entretanto; os valores NÃO podem ser armazenados de forma repetida dentro do vetor.

Link reply.it: [Aqui!](#)

▼ Exemplo 4:

Fazer um algoritmo que compare dois vetores: V1 e V2 e

exibir na tela se os vetores tem os mesmos valores em qualquer ordem.

▼ PESQUISA E ORDENAÇÃO DE DADOS

Apostila: [Aqui!](#)

▼ MÉTODO DE ORDENAÇÃO POR TROCA

Link reply.it: [Aqui!](#)

▼ MÉTODO DE PESQUISA SEQUENCIAL

Link reply.it: [Aqui!](#)

▼ MÉTODO DE PESQUISA BINÁRIA

Link reply.it: [Aqui!](#)

▼ COMO CRIAR VALORES REAIS (FLOAT) ALEATÓRIOS:

Link reply.it: [Aqui!](#)

▼ ARRAY CHAR - STRING

Exemplo de Vetor - CHAR: TEXTO				
POSIÇÃO	1	2	3	4
TEXTO	'U'	'V'	'V'	'\0'
ÍNDICE	0	1	2	3

CARACTERE DE CONTROLE: FIM DA STRING: '\0'

LEITURA / ESCRITA DE VETOR DE CHAR - STRING

Link reply.it: [Aqui!](#)

▼ BIBLIOTECA STRING.H

▼ FUNÇÃO STRLEN:

Conta a quantidade de caracteres de uma string (palavra ou texto) **NÃO** considerando o caractere de controle: '\0'

Link reply.it: [Aqui!](#)

▼ FUNÇÃO STRCMP:

Compara duas strings quaisquer e retorna:

protótipo:

resposta = strcmp(STRING1, STRING2); sendo que a **resposta**:

- **VALOR -1**: A STRING1 é diferente a STRING2: Ordem Alfabética.
- **VALOR 0**: A STRING1 é igual a STRING2.
- **VALOR 1**: A STRING1 é diferente a STRING2: Ordem NÃO Alfabética.

Link reply.it: [Aqui!](#)

▼ FUNÇÃO STRCPY:

Copiar a STRING2 para a STRING1 e

esta função não retorna nada (**VOID**) via comando **return**:

protótipo:

```
void strcpy(String1, String2);
```

Link reply.it: [Aqui!](#)

EXEMPLO - **STRLEN** + **STRCMP** + **STRCPY**: Fazer um programa que leia várias palavras (uma por uma) e

exiba na tela a maior palavra entre todas as lidas do usuário.

PARE (**break**) quando o usuário digitar "SAIR" (*string*).

Link reply.it: [Aqui!](#)

▼ FUNÇÃO STRCAT:

Concatena duas (2) strings, conforme o protótipo:

```
void strcat(String1, String2)
```

RESPOSTA: **STRING1 + STRING2**

Link reply.it: [Aqui!](#)

▼ Diferenças entre CHAR e STRING:

- **CHAR**: Um (1) caractere - **STRING**: Conjunto de caracteres
- **CHAR**: Aspas simples: '@' - **STRING**: Aspas duplas: "@"
- **CHAR**: Caractere da [Tabela ASCII](#): '@' -
- **STRING**: Vetor: "@" =

Exemplo de Vetor - CHAR: ARROBA: @

POSIÇÃO	1	2
TEXTO	'@'	'\0'
ÍNDICE	0	1

- **CHAR**: tipo de variável **CHAR** - **STRING**: Biblioteca na linguagem C: [string.h](#) (header)
- **CHAR**: Um (1) BYTE - **STRING**: Conjunto de BYTE

▼ Contar LETRAS

EXEMPLO: Fazer um programa que leia um texto e uma letra e exiba o número de ocorrências da letra no texto.

Repetir esse processo várias vezes até o usuário digitar "SAIR".

Link reply.it: [Aqui!](#)

▼ Contar SUBSTRINGS:

EXEMPLO: Fazer um programa em que leia um texto qualquer e verifique no texto o número de ocorrências da substring "UVV".

Parar o programa quando a substring NÃO estiver no texto.

Link reply.it: [Aqui!](#)

▼ Diferenciar Minúsculo x Maiúsculo:

Da Tabela ASCII, temos:

- MINÚSCULOS: 'a' (97) até 'z' (122)
- MAIÚSCULOS: 'A' (65) até 'Z' (90)

sendo que, a diferença ASCII entre as letras são sempre de 32 unidades.

EXEMPLO:

Escrever um algoritmo que leia um Texto qualquer e exiba na tela:

- Quantidade de Letras minúsculas;
- Quantidade de Letras maiúsculas;
- Quantidade de demais caracteres;

Repetir esse processo várias vezes até o usuário digitar "SAIR".

OBS: Considerar strings acima de dois (2) símbolos; apenas.

Link reply.it: [Aqui!](#)

▼ Lista de Exercícios - VETORES INT e FLOAT:

Lista: [Aqui](#)

▼ Lista de Exercícios - VETORES DE CHAR: STRINGS

Lista: [Aqui](#)

▼ Referência Bibliográfica:

Site do C - STRING.H: [Link](#)

Na resolução dos exercícios usar **APENAS**:

- **STRLEN**: [Link](#)
- **STRCMP**: [Link](#)
- **STRCPY**: [Link](#)
- **STRCAT**: [Link](#)

▼ \mapsto Curiosidade - ARRAY PYTHON:

Código usado para realizar os sorteios das questões:

```
import numpy as np
vetor = np.random.choice(np.arange(1, 9), 8, replace=False)
vetor

array([2, 6, 8, 7, 1, 3, 5, 4])

vetor = np.zeros((49, 10), dtype=int)
for i in range (49):
    vetor[i] = np.random.choice(np.arange(1, 41), 10, replace=False)
    vetor[i].sort()
vetor

array([[ 2,  3, 11, 12, 20, 24, 25, 29, 34, 40],
       [ 4,  5,  7,  9, 13, 15, 23, 32, 35, 37],
```

```
[ 6, 13, 15, 18, 20, 27, 28, 29, 32, 37],
[ 6, 7, 13, 14, 28, 29, 37, 38, 39, 40],
[ 1, 8, 9, 10, 11, 19, 21, 36, 39, 40],
[ 4, 5, 6, 7, 8, 14, 21, 24, 30, 36],
[ 1, 4, 7, 25, 26, 27, 30, 35, 36, 38],
[19, 22, 23, 27, 28, 30, 32, 35, 36, 39],
[ 1, 4, 12, 15, 16, 18, 23, 30, 35, 36],
[12, 13, 15, 23, 28, 32, 34, 35, 37, 38],
[ 4, 6, 8, 14, 20, 25, 28, 37, 38, 40],
[ 3, 5, 8, 9, 19, 22, 25, 31, 34, 40],
[ 4, 7, 9, 12, 13, 15, 19, 21, 34, 39],
[ 5, 11, 12, 14, 18, 19, 21, 26, 28, 34],
[ 4, 6, 18, 19, 28, 30, 31, 35, 37, 38],
[ 1, 3, 4, 20, 22, 30, 35, 37, 39, 40],
[ 3, 5, 10, 20, 26, 33, 34, 35, 37, 40],
[ 2, 17, 20, 21, 22, 25, 28, 29, 36, 38],
[ 2, 7, 8, 10, 21, 24, 31, 34, 37, 40],
[ 1, 3, 4, 14, 16, 18, 19, 20, 36, 37],
[ 2, 4, 9, 22, 23, 28, 29, 31, 33, 35],
[ 4, 10, 11, 12, 17, 19, 25, 31, 36, 39],
[ 5, 11, 15, 16, 19, 24, 25, 30, 32, 38],
[ 3, 7, 18, 19, 24, 27, 34, 36, 37, 39],
[ 1, 3, 13, 15, 24, 28, 32, 34, 35, 36],
[ 8, 9, 12, 15, 18, 21, 24, 29, 30, 39],
[ 1, 3, 11, 18, 20, 23, 35, 36, 38, 40],
[ 1, 3, 7, 8, 13, 14, 18, 24, 31, 39],
[ 3, 6, 7, 14, 22, 24, 25, 35, 36, 37],
[ 2, 4, 5, 13, 14, 15, 18, 23, 32, 35],
[ 3, 15, 17, 18, 21, 27, 29, 31, 36, 38],
[ 9, 12, 14, 16, 20, 25, 28, 31, 32, 34],
[12, 13, 14, 17, 26, 28, 29, 33, 34, 39],
[ 6, 10, 12, 25, 27, 30, 31, 32, 35, 40],
[ 1, 4, 5, 6, 9, 18, 21, 23, 24, 34],
[ 3, 13, 15, 19, 24, 25, 26, 27, 36, 38],
[ 4, 15, 16, 18, 20, 25, 27, 30, 32, 36],
[ 5, 6, 12, 14, 23, 24, 33, 36, 37, 40],
[ 1, 7, 11, 17, 19, 23, 25, 37, 38, 39],
[ 2, 3, 10, 18, 19, 20, 27, 28, 37, 39],
[17, 20, 22, 24, 27, 29, 31, 35, 37, 40],
[ 9, 11, 12, 14, 17, 30, 34, 38, 39, 40],
[ 6, 8, 10, 11, 15, 18, 21, 33, 36, 39],
[13, 15, 16, 21, 23, 26, 28, 30, 36, 40],
[ 4, 6, 7, 21, 23, 27, 28, 29, 31, 37],
[ 4, 11, 12, 21, 22, 25, 30, 33, 34, 39],
[ 1, 10, 12, 16, 21, 24, 26, 29, 31, 38],
[ 5, 9, 10, 15, 23, 24, 32, 34, 35, 36],
[ 6, 12, 15, 18, 25, 27, 28, 30, 38, 40]])
```

```
import pandas as pd
df = pd.DataFrame(vetor)
df.to_csv('///content/drive/MyDrive/Períodos/2023-2/CC2M/file.csv')
```

```
1001 // 13
```

```
77
```