



# Algoritmos e Programação II

Prof. Joilson dos Reis Brito  
Prof.<sup>a</sup> Noeli A. Pimentel Vaz

# Avaliações da 1a Va

04-09-2023 Atividade Prática Avaliativa 1 (APA1)

26-09-2023 Atividade Prática Avaliativa 2 (APA2)

09-10-2023 Prova

# **VETOR DE CARACTERES**

# Vetor de Caracteres

Conforme estudamos nas aulas anteriores, um vetor é uma estrutura de dados que armazena várias informações do mesmo tipo.

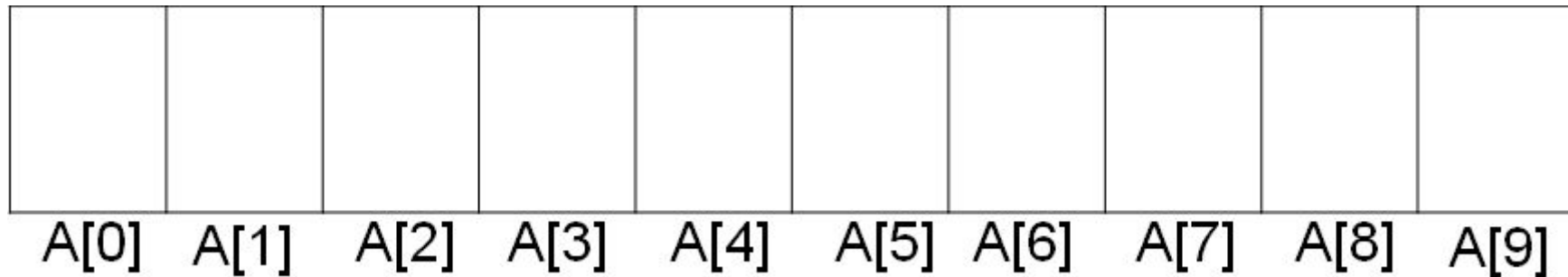
Trabalhamos até agora apenas com vetores que armazenam números, inteiros ou reais.

Agora mostraremos como trabalhar com vetores que armazenam caracteres alfanuméricos

# Vetor de Caracteres

Um vetor de caracteres pode armazenar várias letras, caracteres especiais e números, porém estes números são tratados como letras, se estiverem um vetor de caracteres.

**Exemplo:** `char A[10];`



# Vetor de Caracteres

Um vetor de caracteres pode receber carga inicial:

## Exemplo:

```
#include <stdio.h>
#define TAMANHO 10
int main()
{
    char A[TAMANHO] = {'b','o','r','b','o','l','e','t','a','s'};

}
```

# Vetor de Caracteres

|          |      |      |      |      |      |      |      |      |      |      |
|----------|------|------|------|------|------|------|------|------|------|------|
| <b>A</b> | b    | o    | r    | b    | o    | l    | e    | t    | a    | s    |
|          | A[0] | A[1] | A[2] | A[3] | A[4] | A[5] | A[6] | A[7] | A[8] | A[9] |

## Vetor de Caracteres Leitura

Para ler um vetor de caracteres é necessário colocar o função `fflush(stdin);` antes de cada comando `scanf`.

```
fflush(stdin);  
scanf("%c",&Letras[l]);
```



# Vetor de Caracteres

Um vetor de caracteres pode ser lido:

## Exemplo:

```
#include <stdio.h>
#define TAMANHO 10
int main()
{
    char Letras[TAMANHO];
    int l;
    printf("Informe 10 caracteres:");
    for(l=0;l<TAMANHO;l++)
    {
        fflush(stdin);
        scanf("%c",&Letras[l]);
    }
    for(l=0;l<TAMANHO;l++)
        printf("%c",Letras[l]);

    return 0;
}
```

# Atividade Prática 1

Faça um programa para ler um vetor de caracteres, com 8 caracteres, depois contar quantas letras 'a' existem neste vetor.

# Atividade Prática 1

```
1  #include <stdio.h>
2  #define TAMANHO 8
3  int main()
4  {
5      char Letras[TAMANHO];
6      int I, QuantA = 0;;
7      printf("Informe %d caracteres:", TAMANHO);
8      for(I=0; I<TAMANHO; I++)
9      {
10         fflush(stdin);
11         scanf("%c", &Letras[I]);
12         if(Letras[I] == 'a')
13             QuantA++;
14     }
15     printf("\n%d\n", QuantA);
16     return 0;
17 }
```

# Biblioteca

**Existem bibliotecas específicas para manipulação de caracteres C:**

**Caracteres: ctype.h**

# **BIBLIOTECA PARA MANIPULAÇÃO** **DE CARACTERES ctype.h**

## Biblioteca ctype.h

- A maioria das funções desta biblioteca serve para classificar caracteres ASCII, ou seja, para dizer se um dado byte representa uma **letra**, ou um **dígito**, ou um **branco**, em código ASCII.

# Funções Lógicas - ctype.h

- As funções lógicas abaixo devolvem 0 para dizer “não” ou um inteiro diferente de 0 para dizer “sim”:
  - **isspace(caracter)** → Avalia se o caracter representa um branco;
  - **isdigit(caracter)** → Avalia se o caracter é um dígito decimal (0 1 2 3 4 5 6 7 8 9);
  - **islower(caracter)** → Avalia se o caracter representa uma letra minúscula (a b c d e f g h i j k l m n o p q r s t u v w x y z);

# Funções Lógicas - ctype.h

- As funções lógicas (booleanas) devolvem 0 para dizer “não” ou um inteiro diferente de 0 para dizer “sim”:
  - **isupper(caracter)** → Avalia se o caracter representa uma letra maiúscula (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z);
  - **isalpha(caracter)** → Avalia se o caracter representa uma letra (maiúscula e minúscula);
  - **isalnum(caracter)** → Avalia se o caracter representa um caracter alfanumérico (letra ou dígito decimal);



# Funções Lógicas - ctype.h

- As funções lógicas (booleanas) devolvem 0 para dizer “não” ou um inteiro diferente de 0 para dizer “sim”:
  - **isdigit(caracter)** → Avalia se o caracter representa um caracter numérico (dígito decimal);
  - **ispunct(caracter)** → Avalia se o caracter representa um caracter especial (! " # \$ % & ' ( ) \* + , - . / : ; < = > ? @ [ \ ] ^ \_ ` { | } ~);

# Funções de conversão - ctype.h

- As funções para conversão de caracteres:
  - **toupper(caracter)** → A função toupper recebe uma letra e devolve a letra correspondente maiúscula.
  - **tolower(caracter)** → A função tolower recebe uma letra e devolve a letra correspondente minúscula.

# Resolvendo problemas

As funções da biblioteca ctype nos permitem analisar cada caracter de um vetor e identificar os tipos de caracteres que existem no vetor.

# Atividade Prática 2

Altere o programa para contar e mostrar:

- \_ a quantidade de letras minúsculas
- \_ a quantidade de letras maiúsculas
- \_ a quantidade de caracteres especiais

# Atividade Prática 2

```
S06_exemplo ctype.cpp  [*] S06_Atividade Prática 1.cpp  [*] S06_Atividade Prática 2.cpp
1  #include <stdio.h>
2  #include <ctype.h>
3  #define TAMANHO 8
4  int main()
5  {
6      char Letras[TAMANHO];
7      int I, Minuscula = 0, Maiuscula = 0, CaracEspecial = 0;
8      printf("Informe %d caracteres:", TAMANHO);
9      for(I=0; I<TAMANHO; I++)
10     {
11         fflush(stdin);
12         scanf("%c", &Letras[I]);
13         if(islower(Letras[I]) != 0)
14             Minuscula++;
15         if(isupper(Letras[I]) != 0)
16             Maiuscula++;
17         if(ispunct(Letras[I]) != 0)
18             CaracEspecial++;
19     }
20     printf("\nQuantidade de letras minusculas: %d\n", Minuscula);
21     printf("\nQuantidade de letras maiuscula: %d\n", Maiuscula);
22     printf("\nQuantidade de caracteres especiais: %d\n", CaracEspecial);
23     return 0;
24 }
```

## Atividade Prática 3

Agora vamos considerar que o vetor de caracteres lido é uma senha que um usuário quer cadastrar em um site e que o site só aceita senhas que tenham, pelo menos, 1 letra maiúscula, 1 letra minúscula e um caracter especial.

Altere o programa para não mostrar mais as quantidades e emitir apenas duas mensagens:

Senha Válida ou Senha Inválida.

```
1  #include <stdio.h>
2  #include <ctype.h>
3  #define T 8
4  int main()
5  {
6      char VetChar[T];
7      int X, ContMin = 0, ContMai = 0, ContEsp = 0;
8      printf("Digite %d caracteres: ",T);
9      for(X=0;X<T;X++)
10     {
11         scanf("%c",&VetChar[X]);
12         if(islower(VetChar[X]))//checagem se o caracter é minúsculo
13             ContMin++;
14         if(isupper(VetChar[X]))//checagem se o caracter é maiúsculo
15             ContMai++;
16         if(ispunct(VetChar[X]))//checagem se o caracter é especial
17             ContEsp++;
18     }
19     if(ContMin != 0 && ContMai != 0 && ContEsp !=0)
20         printf("Senha válida.");
21     else
22         printf("Senha inválida.");
23     return 0;
24 }
25 |
```

# EXERCÍCIO

Faça um programa para ler um vetor de 12 caracteres. Após a leitura o programa deve:

- Converter os caracteres que forem letras para maiúsculo.
- Substituir os caracteres que não forem letras pelo asterisco.
- Imprimir o vetor modificado.