



# LABORATÓRIO DE PROGRAMAÇÃO

## Manipulação de Strings em C

Profº. Sérgio Roberto Costa Vieira, M.Sc.  
Cursos de Computação  
1º. Período

# Laboratório de Programação

## Cadeia de Caracteres

- Também conhecida como String, alfanumérica ou literal;
- É uma sequência ordenada de caracteres (símbolos) escolhidos a partir de um conjunto pré-determinado;
- Em programação, cada símbolo armazenado na memória é representado por um valor numérico;
- Esse tipo de literal armazena dados que contêm letras, dígitos e/ou símbolos especiais

# Laboratório de Programação

## Cadeia de Caracteres

- Exemplo: constante *string*
  - “Maria” *string* de comprimento 5
  - “123” *string* de comprimento 3
  - “0” *string* de comprimento 1
  - “A” *string* de comprimento 1



# Laboratório de Programação

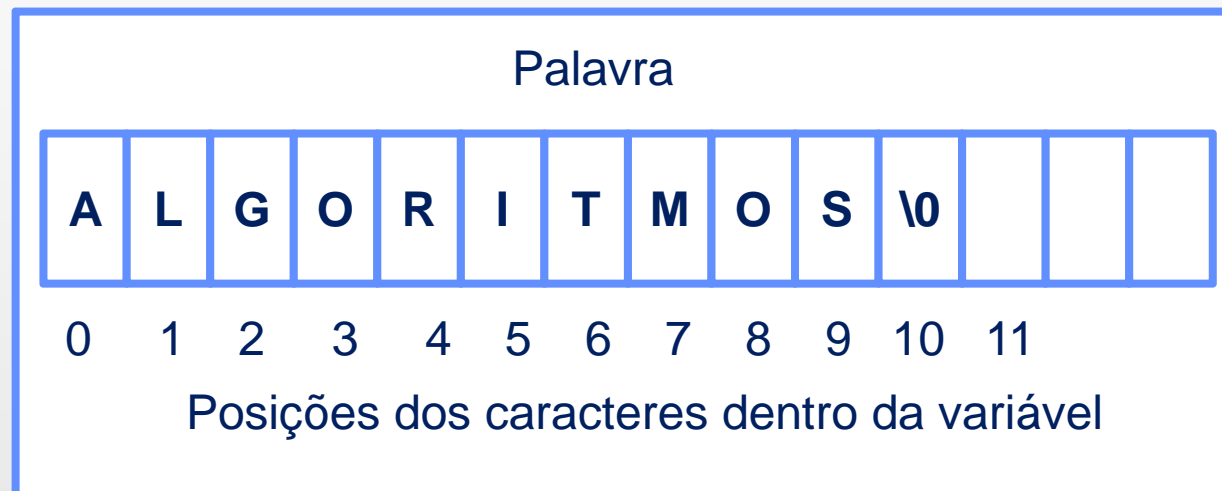
## Cadeia de Caracteres

Suponha que na entrada de dados foi digitado: **ALGORITMOS**

**String** palavra;

**Leia** palavra;

Memória Principal (MP)



*Em algumas linguagens, a numeração poderá começar com 1*

# Laboratório de Programação

## Cadeia de Caracteres

- Não existe um tipo String em C.
- Strings em C são vetores do tipo char que terminam com '\0'.
- Para literais string, o próprio compilador coloca '\0'.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main(){
```

```
    char re[8] = "lagarto"; //tamanho máximo de 7 caracteres
```

```
    printf ("%s", re);
```

```
    system("pause");
```

```
    return(0);
```

```
}
```

# Laboratório de Programação

## Strings

Tipicamente representado na forma de caractere, '\0', ou simplesmente pelo seu valor, 0.

- Para declarar uma *string*, usa-se o seguinte formato
  - `char nome_da_string[tamanho];`
- Exemplo:
  - `char exemplo[4] = {'a','b','c','\0'};`

# Laboratório de Programação

## Strings

Observe que o espaço para o caractere `'\0'` deve ser previsto quando dimensionando o tamanho do arranjo de caracteres que será manipulado como *string*;

C suporta uma forma alternativa de representação de um *string* constante, que é através do uso de aspas;

- Exemplo:

- `char exemplo[4] = "abc";`

`'\0'` é automaticamente anexado à *string* pelo compilador

# Laboratório de Programação

## Para ler uma String

- Comando scanf vs. gets

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    char re[10];
    printf ("Digite o seu nome: ");
    scanf("%s", &re);
    printf ("Oi %s \n", re);
    system("pause");
    return (0);
}
```

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    char re[10];
    printf ("Digite o seu nome: ");
    gets(re);
    printf ("Oi %s \n", re);
    system("pause");
    return (0);
}
```



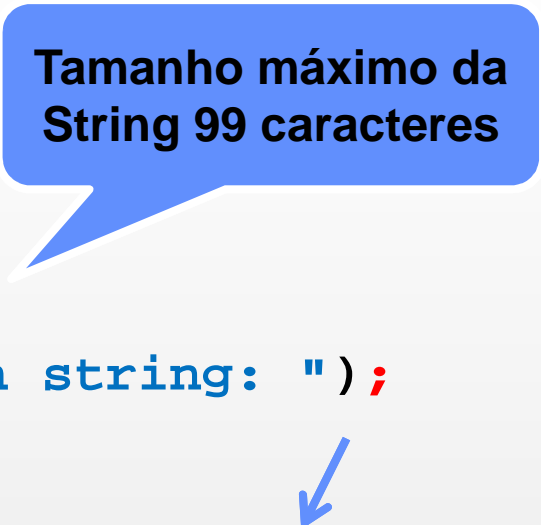
# Laboratório de Programação

## Strings

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    char string[100];

    printf("Digite uma string: ");
    gets(string);
    printf("\n\nVoce digitou %s", string);
    system("pause");
    return(0);
}
```

Tamanho máximo da String 99 caracteres



# Laboratório de Programação

## Strings

```
#include <stdio.h>

#include <stdlib.h>

int main(){

    char str[10] = "Joao";

    printf("\n\nString: %s", str);

    printf("\nSegunda letra: %c", str[1]);

    str[1] = 'U';

    printf("\nAgora a terceira letra e: %c", str[2]);

    printf("\n\nString resultante: %s \n", str);

    system("pause");

    return(0);

}
```

# Laboratório de Programação

## Funções de Manipulação de Strings

- Abaixo as principais funções usadas para a manipulação de *strings* em linguagem C.
- Essas funções estão no arquivo cabeçalho **string.h**

---

•strcpy

•strupr

•strcat

•strlwr

•strlen

•strrev

•strcmp

•strset

**A função strcpy() copia a string-origem para a string- destino**

**Uso:**

```
strcpy (string_destino, string_origem);
```

# Laboratório de Programação

## Função strcpy()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(){
    char str1[100],str2[100],str3[100];
    printf ("Entre com uma string: ");
    gets (str1);
    strcpy (str2,str1);
    strcpy (str3,"Voce digitou a string: ");
    printf ("\n\n %s %s \n",str3,str2);
    system("pause");
    return 0;
}
```



**A *string* de origem permanecerá inalterada e será anexada ao fim da *string* de destino**

**Uso:**

```
strcat (string_destino, string_origem);
```

# Laboratório de Programação

## Função strcat()

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main(){

    char str1[100],str2[100];

    printf ("Entre com uma string: ");

    gets (str1);

    strcpy (str2,"Voce digitou a string ");

    strcat (str2,str1);

    printf ("\n\n%s\n",str2);

    system("pause");

    return 0;

}
```

# Laboratório de Programação

## Para concatenar duas strings

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(){
    char str1[40],str2[10],str3[40]="teste";
    strcpy(str1,"inicio");
    strcpy(str2,"FIM");
    strcat(str1,str2);
    strcat(str3,str1);
    printf("%s\n",str1);
    printf("%s\n",str3);
    system("pause");
    return (0);
}
```

**A função strlen() retorna o comprimento da string fornecida;**

**O terminador nulo não é contado;**

**Uso:**

```
strlen(string);
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(){
    int size;

    char str[100];

    printf("Entre com uma string: ");

    gets(str);

    size=strlen(str);

    printf("\n\nA string que voce digitou tem tamanho %d\n",size);
    system("pause");
    return 0;
}
```



A função strcmp() compara a string 1 com a string 2;

Se as duas forem idênticas a função retorna zero;

Se elas forem diferentes a função retorna:

1 se a string2 for maior

-1 se a string2 for menor

**USO:**

```
strcmp(string1, string2);
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(){
    char str1[100],str2[100];
    printf ("Entre com uma string: ");          gets(str1);
    printf ("\n\nEntre com outra string: ");    gets (str2);

    if (!strcmp(str1,str2))
        printf ("\n\nAs duas strings são diferentes.");
    else
        printf("\n\nAs duas strings são iguais.");
    system("pause");
    return(0);
}
```

# Laboratório de Programação

## Para comparar duas strings

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main( ){
    char re[80];
    printf ("Digite a senha: ");
    gets(re);
    if (strcmp(re,"laranja")==0) {
        printf ("Senha correta\n");
    }
    else {
        printf ("Senha invalida\n");
    }
    system("pause");
    return (0);
}
```

**Funçãostrup() converte uma string para maiúscula**

**USO:**

```
strup(string);
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
    char string[20];
    printf("Entre com a string :");
    scanf("%s",&string);
    printf("Convertendo : %s\n",strupr(string));

    system("pause");
    return(0);
}
```



### Função `strlwr()`;

Converte uma string para minúscula

USO:

```
strlwr(string);
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
    char string[20];
    printf("Entre com a string :");
    scanf("%s",&string);
    printf("Convertendo : %s\n",strlwr(string));
    system("pause");
    return(0);
}
```

**Função `strrev()`;**

**Inverte uma *string* dada**

**USO:**

```
strrev(string) ;
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(){

    char string[20];

    printf("Entre com a string :");

    scanf("%s",&string);

    printf("Invertendo : %s\n",strrev(string));

    system("pause");
    return(0);
}
```

**Substitui todos os caracteres da *string* pelo mesmo caractere usado na função `strset()`**

**USO:**

```
strset(string, char);
```



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(){
    char string[20];
    printf("Entre com a string :");
    scanf("%s",&string);
    printf("Substituindo: %s\n",strset(string, '*'));

    system("pause");
    return(0);
}
```

# Laboratório de Programação

## Convertendo *string* em número

- Estas funções fazem parte do arquivo de cabeçalho `stdlib.h`

FUNÇÃO	CONVERTENDO
<code>atof(string)</code>	<code>float</code>
<code>atoi(string)</code>	<code>int</code>
<code>atol(string)</code>	<code>long int</code>
<code>strtod(string)</code>	<code>double</code>
<code>strtoul(string)</code>	<code>long</code>

# Laboratório de Programação

## Convertendo *string* em número

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    char str[80];
    int resultado;
    float resultado2;

    scanf("%s", &str);
    resultado = atoi(str);
    printf("Conversão:%s-%d\n", str, resultado);

    scanf("%s", &str);
    resultado2 = atof(str);
    printf("Conversão:%s-%f\n", str, resultado2);

    system("pause");
    return 0;
}
```

# Laboratório de Programação

## Exemplo de Uso das Funções

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
    char palavra[30], palavra2[30], teste[30];
    int tam, comp;
    printf ("\n Informe uma string: ");
    gets (palavra);
    printf ("A palavra digitada %c %s\n", 130, palavra);
    printf ("\n Informe uma segunda string: ");
    gets (palavra2);
    printf ("A palavra digitada %c %s\n", 130, palavra2);
    strcpy (teste,"aula teste");
    printf ("A palavra copiada %c %s\n", 130, teste);
```

# Laboratório de Programação

## Exemplo de Uso das Funções

```
strcat (teste, " 1");  
printf ("A palavra concatenada %c %s\n", 130, teste);  
tam=strlen(palavra);  
printf ("O tamanho da 1a. string %c %d",130, tam);  
comp=strcmp (palavra, palavra2);  
if (comp==0)  
    printf ("\nSao iguais: %d", comp);  
    else  
        printf ("\nSao diferentes: %d", comp);  
printf ("\nMaiusculo: %s",strupr(palavra));  
printf ("\nMinusculo: %s\n",strlwr(palavra));  
system("pause");    return (0);  
}
```





# LABORATÓRIO DE PROGRAMAÇÃO

## Manipulação de Strings em C

Profº. Sérgio Roberto Costa Vieira, M.Sc.  
Cursos de Computação  
1º. Período