



3. Linguagem C – Strings e gets

Strings:

- É um vetor de caracteres terminado com um caractere nulo;
 - Devemos apenas ficar atentos para o fato de que as strings têm o seu último elemento como um '\0'.
- * A primeira posição do vetor em C é zero.



Linguagem C

- Forma geral:

char nome_da_string [tamanho];

- Isto declara um vetor de caracteres com o número de posições igual a tamanho. Note, que temos que reservar uma posição para o terminador nulo, ou seja, o tamanho da string tem que ser no mínimo um caractere maior do que o tamanho da string que queremos armazenar.



Linguagem C

- Operações com string: Uma string não é considerado um tipo básico em C (como int, float). Logo, operações como cópia ou comparação não são elementares.

OPERAÇÃO	TIPO BASICO	STRING
Declaração	Int x,y;	Char str[81], aux[81]
Atribuição	X=y;	Strcpy(aux,str)
Comparação	If(x==y)	If(!strcmp(aux,str))

Linguagem C

- Funções para strings
 - **gets()**
 - É utilizada para leitura de uma string através do dispositivo padrão, até que o ENTER seja pressionado.
 - A função gets() não testa os limites na variável que é chamada
 - Forma Geral:


```
gets(nome_variavel);
```

Linguagem C

- Exemplo:

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <locale.h>
int main ()
{
    setlocale(LC_ALL, "Portuguese");
    char string[100];
    fflush(stdin);
    printf ("Digite o seu nome: ");
    gets (string);
    printf ("\n\n Ola %s",string);
    fflush(stdin);
    getch();
}
```



Linguagem C

- Funções para strings

- **strcpy()**

- Copia a string-origem para a string- destino.

- Forma Geral:

strcpy (string_destino,string_origem);



Linguagem C

- Atividade 1: Faça um programa onde o usuário digite uma string. Armazene essa string em uma variável, e copie ela para outra variável. Apresente as duas strings no final.



Linguagem C

Atividade 1:

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <locale.h>
int main ()
{
    char str1[50],str2[50];
    printf ("Entre com uma string: ");
    gets (str1);
    strcpy (str2,str1); /* Copia str1 em str2 */
    printf ("\n\nString Digitada:%s\nString Copiada:%s",str1,str2);
    getch();
}
```



Linguagem C

- Funções para strings

- ***strcat ()***

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

- Forma Geral:

- strcat (string_destino,string_origem);*



Linguagem C

- Atividade 2: Faça um programa onde o usuário digite uma string. Crie outra string com a expressão:

“Você digitou a string: ”.

Concatene as duas strings, de modo que o programa exiba no final a expressão:

Você digitou a string: ‘string digitada pelo usuário’



Linguagem C

Atividade 2:

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <locale.h>
int main ()
{
    setlocale(LC_ALL, "Portuguese");
    char str1[50],str2[50];
    printf ("Entre com uma string: ");
    gets (str1);
    strcpy (str2,"Voce digitou a string ");
    strcat (str2,str1);
    /* str2 armazenara' Voce digitou a string + o conteudo de str1 */
    printf ("\n\n%s",str2);
    getch();
}
```



Linguagem C

- Funções para strings

- **strlen()**

- retorna o comprimento da string fornecida.
 - O terminador nulo não é contado. Isto quer dizer que, de fato, o comprimento do vetor da string deve ser um a mais que o inteiro retornado por strlen().

- Forma Geral:

strlen (string);



Linguagem C

- Atividade 3: Faça um programa onde o usuário digite duas strings. Utilize a função *strlen()* para contar quantos caracteres existem em cada string e apresente no final qual string é maior. Informe se são iguais no tamanho.



Linguagem C

```
Atividade 3:
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <locale.h>
int main ()
{
    setlocale(LC_ALL, "Portuguese");
    int tamanho1,tamanho2;
    char str[100],str2[100];
    printf ("Entre com uma string: ");
    fflush(stdin);
    gets (str);
    printf ("Entre com outra string: ");
    gets (str2);
    tamanho1=strlen (str);
    tamanho2=strlen (str2);
    if(tamanho1>tamanho2){
        printf ("\nA string 1 %s é maior. Possui tamanho %d",str,tamanho1);
    }else if (tamanho2>tamanho1){
        printf ("\nA string 2 %s é maior. Possui tamanho %d",str2,tamanho2);
    } else{
        printf ("\nAmbas as strings (%s e %s) tem o mesmo tamanho %d e %d",str,str2,tamanho1,tamanho2);
    }
    getch(); // força uma parada do programa no final da sua execução.
}
```



Linguagem C

- Funções para strings
 - ***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 não zero.
 - Forma Geral:
strcmp (string1,string2);



Linguagem C

- Atividade 4: Faça um programa onde o usuário digite duas strings. Utilize a função *strcmp()* para verificar se elas são idênticas. Informe no final com mensagem adequada.



Linguagem C

- Atividade 4:

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <locale.h>
int main ()
{
    setlocale(LC_ALL, "Portuguese");
    char str1[100],str2[100];
    printf ("Entre com uma string: ");
    fflush(stdin);
    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.");
    getch(); // força uma parada do programa no final da sua execução.
}
```